وزارت علوم، تحقیقات و فناوری

مرکز منطقه ای اطلاع رسانی علوم و فناوری

گزارش نهایی طرح پژوهشی

بکارگیری سیستمهای NOSQL به عنوان ابزار پردازش کلان داده ها

در جستجو و بازیابی تمام متن

Using NoSQL Systems as Big Data Processing Tools
for Full-Text Search and Retrieval

مجری : دکتر بهاره پهلوان زاده

مهر ۱۳۹۸

بسم الله الرحمن الرحيم

چکیده

امروزه داده ها به عنوان یک دارائی ملی شناخته می شوند. همچنین پردازش، تحلیل و استفاده از داده به عنوان یک عامل کلیدی برای رشد کلیه سازمانها تبدیل شده است و موجب مزیت رقابتی در کسب و کارها، محرک نوآوری، افزایش رقابت و اثرات مثبت اجتماعی خواهد شد. مرکز منطقه ای اطلاع رسانی علوم و فناوری نیز از این قضیه مستثنی نمی باشد و با توجه به افزایش حجم داده ها بصورت تصاعدی (برای مثال در طی سال اخیر رشد ۵۰٪ داده های آن را بر بستر پایگاه داده رابطه ای Microsoft SQL Server شاهد بودیم) و همچنین با لحاظ نمودن ماموریتهای این مرکز در ارائه خدمات از طریق پایگاههای داده مقالات تمام متن و موتور جستجوی اختصاصی آن، نیاز به ارائه راهکارهای نوین پردازشی یک ضرورت محسوب میشود.

در این پژوهش با توجه به ورود تکنولوژیهای نوینی همچون پایگاههای داده غیر رابطه ای و سیستمهای NOSQL در عصر داده های حجیم، ضمن کسب دانش کار با سیستمهای نوظهور NOSQL به بررسی بکارگیری و ارزیابی آنها در قیاس با پایگاه داده رابطه ای موجود بعنوان راهکاری احتمالی برای حل مشکلات آتی و پیش رو در اثر رشد تصاعدی داده های مبتنی بر سند مرکز منطقه ای پرداخته شده است.

برای ارزیابی مزایا و معایب پایگاههای داده غیر رابطه ای در قیاس با پایگاههای داده رابطه ای، معیار زمان بازیابی و معیار جدید کیفیت بازیابی تعریف گردید و پایگاه های داده رابطه ای Microsoft SQL و MariaDB و پایگاه های داده غیر رابطه ای MongoDB و Elasticsearch مقایسه گردیدند. نتایج پژوهش برتری چشمگیر ElasticSearch و Microsoft SQL Server نسبت به دو پایگاه داده دیگر از دیدگاه معیارهای زمان بازیابی و کیفیت بازیابی را نشان داد. همچنین مشاهده شد که با افزایش تعداد شاردها در Elastic Search؛ Elasticsearch برتری خود را نسبت به سایر پایگاههای داده افزایش می دهد.

کلید واژه ها: پردازش و تحلیل داده ها، کلان داده ها، پایگاه داده رابطه ای ، پایگاه داده غیر رابطه ای (سیستمهای NOSQL)

فهرست مطالب

فهرست جداول

فهرست شکل ها

فصل اول : مقدمه

۱–۱ مقدمه

همانطور که می دانیم در عصر داده های حجیم؛ پردازش، تحلیل و استفاده از داده به عنوان یک عامل کلیدی برای رشد کلیه سازمانها تبدیل شده است و موجب مزیت رقابتی در کسب و کارها، محرک نوآوری، افزایش رقابت و اثرات مثبت اجتماعی خواهد شد. بر اساس گزارشهای سالانه سایت IEEE [۱] بعنوان بزرگترین انجمن تخصصی دنیا در پیشرفت تکنولوژی و استاندارد ها برخی از رایجترین و محبوبترین واژه های جستجو شده شامل موارد زیر است:

- کلان داده ها[۲]
- اینترنت اشیا[۳]
- امنیت سایبری[۴]
- رایانش ابری[۵]
- نسل جدید شبکه های بی سیم[۶]
- شبکه های مشبک توری هوشمند[۷]

۱ Institute of Electrical and Electronics Engineers

۲ Big Data

۳ Internet of things

۴ Cyber Security

۵ Cloud Computing

۶ Next Generation Wireless

۷ Smart Grid

از طرفی با توجه به چشم اندازهای آینده و رو به رشد مرکز منطقه ای و متناسب با سند راهبردی و همچنین اساسنامه آن استفاده از شیوه های نوین پردازش برای ارائه برنامه ها و خدمات اطلاع رسانی در منطقه تاکید شده است. با توجه به افزایش روزافزون حجم بزرگ داده ها، گسترش کیفی و کمی سرویسهای متعدد ارائه شده به سایر پژوهشگران در سطح ملی و بعضا بین المللی منطقه؛ حرکت به سمت ارتقا زیرساختهای نرم افزاری در قالب محاسبات توزیع شده و استفاده از روشهای نوین پردازش داده ها اهمیت می‌یابد.

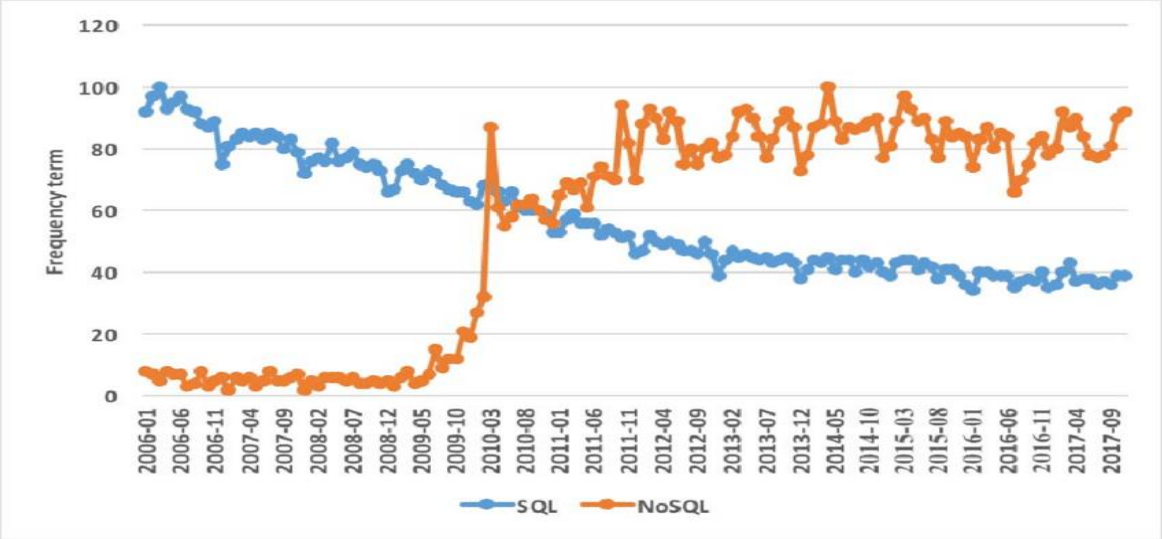
در این پژوهش هدف اصلی کار بر نمونه چکیده های لاتین از پایگاههای اطلاعاتی مرکز به عنوان داده های مورد پژوهش و بررسی می باشد. ضمن اینکه از داده های در حال حاضر حدود ۶میلیون رکورد چکیده های لاتین در دست است (که در سال اخیر رشد چشمگیر بالغ بر ۵۰٪ی آن را شاهد بودیم)، که با توجه به خصوصیت جستجوی تمام متنی[1] ، هر رکورد به صورت تقریبی حدود ۳۰۰ کلمه است و لذا داده های مورد بررسی بالغ بر یک میلیارد و هشتصد میلیون می باشد که به نوبه خود داده حجیم محسوب می شود.

بنابراین انجام این طرح باعث کسب دانش کار با سیستمها و پایگاههای داده غیررابطه ای NOSQL[2] و حرکت به سمت خوشه بندی داده ها در مرکز در سالهای آتی می شود. از طرفی استفاده از این دانش نه تنها در حجم بالای داده باعث بهبود سرعت بازیابی می‌شود بلکه برای ذخیره سازی داده های حجیم در آینده نزدیک برای مرکز ضروری است، چرا که این روزها با وجود چنین تکنولوژیهایی هزینه های مقیاس پذیری عمودی[3] غیر قابل توجیه و بعضا تحمل ناپذیر بوده و باید با

[1] Full-Text Search
[2] Not Only SQL (Structured Query Language)
[3] Vertical Scalability

استفاده از تکنیک های جدید کلاسترینگ در پایگاههای داده غیررابطه ای یا سیستمهای NOSQL، پایگاه داده ها به سمت معماریها و سیاستهای مقیاس پذیری افقی[1] بروند.

پایگاه های داده غیر رابطه ای NOSQL برای کار با داده های حجیم به صورت توزیع شده طراحی شده‌اند. می‌توان با بکارگیری تعداد زیادی گره[2] و توزیع داده‌ها و محاسبات بین آن‌ها از ویژگی مقیاس و گسترش‌پذیری و توسعه منابع در آن‌ها بهره برد. باید در نظر داشت که توزیع خودکار داده و میزان تحمل خطای بالا از ویژگی‌های بارز پایگاه‌های داده غیر رابطه ای است. در شکل ۱-۱ نمودارگوگل ترند[3] رشد استفاده و تحقیقات در زمینه پایگاه های داده غیر رابطه ای NOSQL در قیاس با پایگاه های داده ای رابطه ای را نشان می‌دهد.



شکل ۱-۱ میزان رشد تحقیقات درزمینه پایگاه های داده غیررابطه ای درمقایسه با پایگاهای داده رابطه ای(Google Trends, 2019)

[1] Horizontal Scalability
[2] Node
[3] Google Trends (https://trends.google.com/)

علاوه بر دلایل مطرح شده در بالا و با توجه به اینکه تا به امروز امکان جستجو و بازیابی تمام متن در نرم افزارهای استفاده شده در مرکز منطقه ای به صورت کارا وجود نداشته است، لذا در طرح فعلی برای بهبود کیفیت نتایج جستجو و بازیابی تمام متن به بررسی پایگاه داده های غیر رابطه ای یا همان سیستمهای NOSQL به عنوان روشی نوین برای پردازش داده ها پرداخته می شود. با بررسی انجام شده، بیش از ۲۲۵ نوع سیستم NOSQL وجود دارد که باید با درنظر گرفتن پارامترهایی مانند ساختار ذخیره سازی داده، سرعت خواندن/نوشتن و مکانیزم‌های بهینه سازی آن‌ها متناسب با نوع داده ها و نیازهای مرکز منطقه ای در این طرح پژوهشی محبوب‌ترینها را منتخب و در قیاس با پایگاه داده رابطه ای موجود در سازمان پرداخت.

۱-۲ ضرورت انجام طرح پژوهشی و اهداف کاربردی آن

نیاز عمده نرم افزارهای نسل جدید کتابخانه ای دیجیتال مبتنی بر اینترنت اشیاء جهت ارائه سرویسهای هوشمندانه و کاربرپسند در کتابخانه های مدرن ، حجم بالای داده ها و رشد سریع داده های تمام متن مرکز منطقه ای در سالیان اخیر، همچنین ضرورت استفاده از تکنیکهای خوشه بندی و سازگاری با ذخیره سازی داده ای غیر ساخت یافته (در پایگاههای غیر رابطه ای) با لحاظ نمودن سرعت پردازشی بهتر به عنوان راهکارهای مقیاس پذیر در راستای مدیریت بهینه منابع سخت افزاری از دیگر مواردی است که انجام پژوهش و تحلیل بر داده های موجود در مرکز منطقه ای را با استفاده از روشهای پردازش نوین ضرورت می بخشد. از اینرو است که در برنامه راهبردی نیز به ضرورت استفاده از پردازشهای نوین بر داده های رو به رشد مرکز تاکید به عمل آمده است؛ لذا این پژوهش شروعی در راستای یکی از اهداف برنامه راهبردی مرکز منطقه ای می باشد.

با توجه به وجود چالشهای متفاوت در حوزه کلان داده ها در سطح جهان و ایران به عنوان

کشوری در حال توسعه و عدم وجود بلوغ کافی در حوزه کلان داده در اکوسیستم جهانی و ازطرفی با توجه به نقشه راه توسعه کلان داده کشور (تدوین شده در دی ماه ۱۳۹۶) در جهت ارائه مدلهای جدید برای حل مسائل داخلی و آسیب شناسی توسعه فناوری در کشور از طریق شناخت تجربیات و الگوهای موفق و نا موفق به منظور بهره برداری در حوزه کلان داده ها ؛ و نیز با هدف ایجاد ساز و کاری جهت اندازه گیری میزان ارزش استخراج شده از داده های موجود در کشور با روشهای سنتی در مقابل روشهای نوین پیشنهادی؛ بر آن شدیم که با توجه به رشد نسبتا زیاد داده های موجود در پایگاههای داده مرکز منطقه ای، ضمن بررسی بنیادین راهکارهای نوین پردازشی در این حوزه به پیاده سازی و مقایسه مدلهای انتخابی پرداخته و در صورت کسب نتایج مثبت در راستای سیاستهای آتی مرکز جهت پردازش داده های موجود در پایگاه داده های موجود استفاده نماییم.

مسلما سرمایه گذاری اولیه در این زمینه می تواند گام موثری در راستای شناخت نیازهای واقعی و ایجاد زمینه های پژوهشی و برنامه ریزی برای تولید محصولات در راستای ارائه خدمات بهتر باشد.

۱–۳ سئوالات پژوهش

در انجام این طرح سئوالات پژوهش زیر مطرح می‌باشد:

- آیا با توجه به حجم داده ها و رشد آتی آنها پایگاههای غیر رابطه ای/سیستمهای NOSQL انتخاب مناسبی در راستای برآورده کردن نیازهای مرکز منطقه ای می باشد؟
- کدامیک از پایگاههای غیر رابطه ای/سیستمهای NOSQL بهترین تناسب را با اهداف کاربردی مطرح شده در این طرح دارد؟

۱–۴ روش پژوهش

بصورت اجمالی، مراحل پژوهش و نقشه راه طرح مزبور متناسب با هدفهای ذکر شده و فرضیه های موجود در جهت دانش افزایی مضاعف در این حوزه با توجه به گسترش روز افزون داده ها (برای برنامه های توسعه زیرساختی آتی سازمانی) بشرح زیر است:

- بررسی مدلهای پایگاه‌های داده غیر رابطه ای / سیستمهای NoSQL و انتخاب دو مدل برگزیده
- مدل‌سازی سخت‌افزاری مورد نیاز در حالت آزمایشگاهی
- نصب و راه اندازی بستر سخت افزاری جهت راه‌اندازی معماریهای مختلف تک نود و خوشه ای و انجام تنظیمات مورد نیاز
- نصب و راه اندازی بستر نرم افزاری و انجام تنظیمات پایگاه‌های داده غیر رابطه ای/سیستمهای NoSQL
- بررسی و اجرای داده های معیار[1] بر روی هر یک از پایگاه‌های داده غیر رابطه ای/سیستمهای NoSQL و تحلیل و بررسی نتایج
- نمونه‌گیری از پایگاه داده‌های متنی مرکز منطقه‌ای
- درج اطلاعات در مدل‌های پایگاههای داده غیر رابطه ای انتخاب شده
- اجرا پرس و جوهای آزمایشی[2] جهت بازیابی اطلاعات در محیط توزیع شده در پایگاه‌های داده غیر رابطه ای/سیستمهای NoSQL انتخاب شده
- درج اطلاعات در مدل پایگاه‌های داده رابطه‌ای

[1] Benchmark Data
[2] Query

- اجرا پرس و جوهای آزمایشی[1] جهت بازیابی اطلاعات در مدل پایگاه‌های داده رابطه-ای
- بررسی نتایج حاصل شده و نتیجه‌گیری

۱–۵ هم راستایی طرح پژوهشی حاضر با اهداف سند راهبردی مرکز منطقه‌ای

- به طور کلی : متناسب با اساسنامه و سند راهبردی در راستای هدف استفاده ازتکنولوژی ها و شیوه های نوین پردازش برای ارائه برنامه ها و خدمات اطلاع رسانی در منطقه
- به طور خاص : متناسب با اهداف برنامه های شماره ۳ و۴ و ۵ جدول ۱ "برنامه یک ساله و پنج ساله نظام جامع رایانه ای مرکز" درج شده در فصل هفتم برنامه راهبردی (سند راهبردی مرکز منطقه ای اطلاع رسانی علوم و فناوری، ۱۳۹۵)

۱–۶ خروجی‌های طرح پژوهشی

خروجی این طرح پژوهشی متناسب با اهداف و فرضیه های مطرح شده در قسمت۳-۱ می باشد. و به عبارت دیگر بطورکلی هدف پژوهش و بررسی سیستمهای NoSQL در پایگاههای داده متنی در مقایسه با پایگاههای داده رابطه ای Microsoft SQL Server و Maria DB و نهایتا مقایسه و انتخاب مدل (های) برگزیده مناسب در این حوزه پژوهشی با دیدگاه زمان بازیابی و کیفیت بازیابی برای داده های مرکز منطقه ای اطلاع رسانی علوم و فناوری می باشد. کما اینکه پژوهش در این زمینه منجر به ایجاد دیدگاه تخصصی تر و دانش افزایی دوچندان در حوزه نوین پردازش داده های حجیم سازمانی می‌گردد و در صورت استفاده از نتایج آن در راستای سیاستهای آتی در خصوص زیرساختهای نرم

[1] Query

افزاری پایگاههای داده مرکز به عنوان CoreBusiness سازمان ودر نتیجه رقابت پذیری با رقبا موثر خواهد بود.

۱–۷ استفاده نتایج طرح پژوهشی در سازمان‌ها

هدف اولیه و اصلی مرکز منطقه ای (بصورت نمونه و مطالعه مورردی)[1] است و جامعه آکادمیک نیز از نتایج این بررسی با توجه به نوین و عملیاتی بودن کار میتوانند استفاده ببرند. همانطور که در ضرورت پژوهش عنوان شد،

علاوه بر مرکز منطقه ای، با توجه به نوین بودن پژوهش طرح حاضر هر سازمان دیگری همچون آن که دارای حجم زیادی پایگاه داده متنی می باشد در سطح ملی و بین المللی از نتایج این طرح استفاده نماید. از آن جمله می توان به ISC یا ایران داک با ماموریت های مشابه با مرکز منطقه ای اشاره نمود.

[1] Case Study

فصل دوم : پیشینه پژوهش

۲–۱ مقدمه

پایگاه‌های داده رابطه ای از سال ۱۹۷۰ تا به امروز در حال توسعه اند (Chen, J. K. et al., 2019). استفاده و مدیریت این پایگاه‌های داده با استفاده از سیستم های مدیریت پایگاه داده رابطه ای[1] به سادگی امکان پذیر است و همین امر منجر به محبوبیت این پایگاه‌های داده شده است. با این وجود، با توجه به استفاده روزافزون از فناوری هایی که با تولید اطلاعات در ابعاد و انواع مختلف همراه هستند، سازمانها نیاز به ذخیره سازی حجم بالاتری از اطلاعات نسبت به گذشته دارند. همچنین نیاز است که بازیابی این اطلاعات در سریع ترین زمان ممکن انجام شود.

پردازش این حجم بالای اطلاعات نیازمند سرعت کافی، ساختار انعطاف پذیر و پایگاه‌های داده توزیع شده می‌باشد و پایگاه‌های داده غیر رابطه ای یا همان سیستمهای NOSQL برای پاسخگویی به این نیاز ها طراحی شده اند.

در نقطه مقابل پایگاه‌های داده رابطه ای، برای بازیابی اطلاعات پیچیده در بسیاری از مواقع نیاز به انجام SQL Join بین تعداد دو یا بیشتر جدول دارد که ممکن است باعث افت عملکرد عملیات شود. همچنین عدم سازگاری این نوع پایگاه‌های داده با انبوه اطلاعات بدون ساختار مشخص و نداشتن قابلیت توزیع پذیری از دیگر نقاط منفی آن ها نسبت به پایگاه‌های داده غیر رابطه ای

[1] RDBMS (Relational Database Management System)

می‌باشد(Chen, J. K. et al., 2019; Li, Y, et al., 2013). در ادامه به توضیح مفصل تری از مفاهیم ارائه شده در این قسمت می پردازیم.

۲-۲ پایگاه‌های داده رابطه‌ای

پایگاه داده رابطه‌ای نوعی از پایگاه داده است که داده‌های مرتبط به هم را ذخیره کرده و دسترسی به آن ها را فراهم می آورد. به عبارتی این نوع پایگاه‌های داده مبتنی بر مدل رابطه ای[1] می باشند که یک راه شهودی و قابل درک برای نشان دادن اطلاعات در جداول است. در این نوع پایگاه داده هر سطر یک جدول، یک رکورد با یک کلید شناسایی منحصر به فرد است و هر ستون نمایانگر یک ویژگی در مورد رکورد های آن جدول است(Oracle-RDBMS, 2017).

۲-۲-۱ مدل رابطه ای

در سال‌های اولیه توسعه پایگاه‌های داده، هر برنامه کاربردی[2] داده‌های خود را در ساختار منحصر فرد خود ذخیره می کرد و همین امر باعث می شد توسعه این برنامه‌ها و برنامه های دیگر با استفاده از این داده‌ها، نیازمند مطلع شدن از جزئیات آن ساختار داده خاص باشد. اینگونه ساختارهای داده ناکارآمد بودند و بهینه سازی و ارتقای آن‌ها دشوار بود. برای حل این مشکل مدل پایگاه داده رابطه‌ای طراحی و به عنوان جایگزینی برای ساختارهای داده دلخواه پیشین مطرح شد.

مدل رابطه‌ای روشی استاندارد برای نمایش داده‌ها و جستجو در آن‌ها فراهم می آورد که برای هر برنامه‌ای قابل استفاده می‌باشد. از همان ابتدا، توسعه دهندگان دریافتند که قدرت اصلی این

[1] Relational
[2] Application

مدل، در استفاده از مفهوم جدول است که روشی بصری، کارآمد و تا حدودی انعطاف پذیر برای ذخیره و دسترسی به اطلاعات ساختاریافته است.

با گذشت زمان، قدرت دیگری از مدل رابطه‌ای پدیدار شد و آن استفاده از زبان جستجوی ساختاریافته SQL برای ارتباط با پایگاه‌های داده رابطه‌ای بود. این زبان بر اساس جبر رابطه‌ای، یک زبان تقریبا ریاضی در اختیار کاربران قرار می دهد و به بهبود عملکرد پرس و جوهای آزمایشی بر روی پایگاه‌داده کمک شایانی می کند.

۲-۲-۲ سیستم های مدیریت پایگاه‌های داده رابطه‌ای

یک سیستم مدیریت پایگاه داده رابطه‌ای، مجموعه‌ای از برنامه‌ها و قابلیت‌هایی است که تیم های IT و سایر کاربران را قادر می سازد به ایجاد یک پایگاه‌داده رابطه‌ای، مدیریت و ارتباط با آن بپردازند. بیشتر سیستم های مدیریت پایگاه‌های داده رابطه‌ای های تجاری برای دسترسی به پایگاه‌داده زبان SQL[1] استفاده می‌کنند (Gudivada, V. N. et .al, 2014) از محبوب ترین این سیستم ها، سیستم‌های مدیریت پایگاه داده Microsoft SQL Serverو MySQLمی باشند که هر دو به زبان های C و C++ توسعه داده شده اند و اولین نسخه آن‌ها به ترتیب در سال‌های ۱۹۸۸ و ۱۹۹۵ عرضه شد (DB-Engines Ranking, 2017) . پس از انتقال مالکیت کامل پروژه MySQL از شرکت MySQL ABبه شرکت Oracle در بین سال های ۲۰۰۸ و ۲۰۰۹، توسعه دهندگان اصلیMySQL ، پایگاه داده

[1] Structured Language Query

MariaDBرا به عنوان یک شاخه و انشعاب[1] از MySQL معرفی کردند و تا به امروز این پایگاه‌داده نوپا در بسیاری از شرکت ها جایگزین MySQL شده است(Kenler, E., 2015).

۲-۳ پایگاه‌های داده غیر رابطه ای

از اینکتومی [2]به عنوان اولین موتور جستجوی واقعی تا گوگل رهبر فعلی موتورهای جستجو، پژوهشگران و توسعه دهندگان به خوبی محدودیت‌های پایگاه‌های داده رابطه‌ای را دریافتند. از جمله این محدودیت‌ها که بیشتر در مقیاس بزرگ اطلاعات خود را نشان می دهند، مقیاس پذیری، موازی سازی عملیات و هزینه است (Vaish, G. 2013). پژوهش هایی از این دست و موفقیت اولین پایگاه‌های داده غیررابطه‌ای جدول بزرگ گوگل[3] و پایگاه داده داینامو آمازون[4] (Amazon DynamoDB,2017) و ، آغازگر توسعه پایگاه‌های داده غیر رابطه‌ای متعددی بود که بصورت متن باز عرضه شدند (Li, Y. et al., 2013). این پایگاه‌های داده دارای ویژگی های اصلی زیر می باشند که باعث افزایش روزافزون محبوبیت آن‌ها شده است(Vaish, G. 2013).

- ذخیره و بازیابی داده‌های بدون ساختار در قالب فرمت های مختلف
- افزایش سرعت توسعه برنامه‌های مختلف چرا که برخلاف پایگاه‌داده های رابطه‌ای دسترسی به داده‌های مرتبط نیازی به استفاده از پرس و جوهای آزمایشی پیچیده‌ی SQL ندارد.
- افزایش سرعت دسترسی به داده‌های مختلف

[1] Fork
[2] Inktomi
[3] Google's BigTable
[4] Amazon's Dynamo

- قابلیت مقیاس پذیری افقی به معنای اینکه بتوانیم به منظور بهبود عملکرد، بجای اینکه سخت افزار یک سیستم واحد را ارتقاء دهیم (مقیاس پذیری عمودی)، از سیستم های متعددی در قالب یک سیستم توزیع شده بهره مند شویم.

۲-۳-۱ طبقه بندی پایگاه‌های داده غیر رابطه ای NOSQL

پایگاه‌های داده غیر رابطه‌ای NOSQL بر اساس ساختار ذخیره سازی داده ها بصورت زیر به چهار دسته کلی قابل طبقه بندی می باشد (Han, J., et al., 2011; Abramova, V. et al., 2014; Patil, M. M., 2017) که در ادامه به شرح هریک از آن‌ها بصورت اجمالی خواهیم پرداخت. جدول ۲-۱ نیز به مقایسه اجمالی ۴ دسته پایگاه داده غیر رابطه ای می‌پردازد.

- پایگاه داده کلید-مقدار[1]
- پایگاه داده مبتنی بر ستون[2]
- پایگاه داده سند گرا[3]
- پایگاه داده مبتنی برگراف[4]

[1] Key-Value Database
[2] Column oriented
[3] Document Based
[4] Graph Based

جدول ۲-۱مقایسه چهار دسته پایگاه داده غیر رابطه ای بر حسب مدل داده

مدل داده	کارآیی	مقیاس پذیری	انعطاف پذیری	پیچیدگی	عملکرد
کلید-مقدار	بالا	بالا	بالا	-	متغیر
مبتنی بر ستون	بالا	بالا	متوسط	پایین	حداقل
سند گرا	بالا	متغیر	بالا	پایین	متغیر
مبتنی بر گراف	متغیر	متغیر	بالا	بالا	متغیر

۲-۳-۱-۱ پایگاه داده کلید-مقدار

پایگاه داده کلید-مقدار ساده ترین مدل پایگاه داده غیر رابطه‌ای از لحاظ پیاده سازی می‌باشد. این مدل از یک جدول هاش[1] استفاده می کند و به هر رکورد یک کلید منحصر به فرد و یک اشاره گر[2] به داده اصلی اختصاص می یابد. به عبارتی دیگر، این پایگاه داده‌ها را می‌توان به عنوان یک جدول هاش توزیع شده [3]یا دیکشنری بزرگ در نظر گرفت که هر داده (مقدار) با یک کلید منحصر به فرد آدرس‌دهی می‌شود. داده‌ها از یکدیگر مستقل بوده و ارتباطات بین آن‌ها در سطح برنامه کاربردی مدیریت می‌شوند. درج داده در آن‌ها به راحتی انجام می‌شود. هیچ رابطه‌ای در این

[1] Hash Table
[2] Pointer
[3] Distributed Hash table

ساختار وجود ندارد و برای ذخیره سازی سریع داده‌های پایه استفاده می‌شود (مانند ذخیره سازی نام کاربری و رمز عبور،اطلاعات سژن/نشست[1] ، سوییچینگ[2] و . . .). گروه‌بندی کلیدها و درج مقادیر با ساختارهای تودرتو ازجمله روش‌های ذخیره سازی داده‌های پیچیده در این نوع پایگاه داده می‌باشد (R. Hecht, S. Jablonski, 2011)

مهمترین ویژگی عملیاتی این توع از پایگاه داده‌ها سرعت پاسخ‌دهی برخط[3]و مدیریت حجم بالایی از داده با قابلیت مقیاس‌پذیری خوب بین گره‌های یک خوشه می‌باشد. شعار بانک اطلاعاتی کلید-مقدار آئرواسپاک [4]در این گروه قابل تامل است که ادعا می کنند ۹۹ درصد از درخواست‌ها را در کمتر از ۱ میلی‌ثانیه پاسخ می‌دهد. مثال های این نوع پایگاه‌داده غیر رابطه‌ای پایگاه داده غیررابطه ای اوراکل[5] و پایگاه داده ساده آمازون [6](Amazon SimpleDB, 2017)و ردیس[7] (Macedo, T., & Oliveira, F. (2011)، ولدمورت[8] (Voldmort Project,2017)،ریاک[9] (Riak, 2017)، پایگاه داده آمازون داینامو (DynamoDB,2017) می باشند.

۲-۳-۱-۲پایگاه داده مبتنی بر ستون

پایگاه داده مبتنی بر ستون همانطور که از نامش پیداست، داده‌ها را در قالب ستون، به‌جای سطر، ذخیره می‌کند. این مدل به منظور پردازش حجم بالای اطلاعات توزیع شده روی سرورهای متعدد طراحی شده است. در این مدل ذخیره داده روی یک نقشه مرتب شده توزیع شده[10] و

[1] Session information
[2] Swithcing
[3] Real-Time
[4] AeroSpike
[5] Oracle NoSQL Database
[6] Amazon simpleDB
[7] Redis
[8] Voldemort
[9] RIAK
[10] Distributed Sorted-map

چندبعدی صورت می گیرد. از نظر مفهومی، یک پایگاه داده مبتنی بر ستون مانند یک سیستم های مدیریت پایگاه‌های داده رابطه‌ای با شاخص روی هر ستون بدون سربار مرتبط است. این پایگاه ها، برای اولین بار با ارائه جدول بزرگ گوگل(Chang, F., 2008) و معرفی "سیستم ذخیره سازی توزیع شده برای مقیاس پذیری داده‌های غیر ساخت‌یافته در اندازه بسیار بزرگ" آغاز به کار کردند و بعد از آن نسخه‌های مختلفی از پایگاه داده‌های مبتنی بر ستون ایجاد شدند. ستون‌ها در دسته‌های بزرگتری به نام "خانواده ستون‌ها"[1] دسته‌بندی می‌شوند. انعطاف‌پذیری بالایی در افزودن سطرها و ستون‌ها وجود دارد اما خانواده ستون‌ها باید از پیش تعریف شده باشد. زمانی که یک مقدار تغییر می‌یابد، مقدار جدید به عنوان نسخه متفاوتی از مقدار قبلی با یک مهر زمان ذخیره میگردد. به بیان دیگر مفهوم به روزرسانی به طور مؤثر وجود ندارد .پایگاه داده مبتنی بر ستون، عملیات تجمیع مانند محاسبه ماکزیمم، مینیمم، میانگین و جمع را روی پایگاه داده وسیع با بهترین راندمان انجام میدهند.

پایگاههای داده غیر رابطه ای اچ-بیس[2] (HBase, 2017)، جدول هایپر[3] (Hypertable, 2017) و کاساندرا[4] (Carpenter, J., and Hewitt, E., 2020) مثال هایی از این مدل هستند. پایگاه‌های داده اچ-بیس و جدول هایپر بر مبنای جدول بزرگ بنا نهاده شده‌اند اما پایگاه داده کاساندرا ساختار متفاوتی داشته و در آن مفهومی بنام "فوق ستون"[5] که شامل چندین خانواده ستون می‌باشد، وجود دارد؛ بنابراین برای ذخیره‌سازی داده‌های پیچیده و معنایی مناسبتر می‌باشد. یکی از تفاوت‌های پایگاه داده های رابطه‌ای در مقایسه با پایگاه داده های مبتنی بر ستون نحوه ذخیره سازی مقدار خالی نول[6] می‌باشد. در پایگاه‌دادهای رابطه‌ای باید به ازای هر صفت در یک سطر در صورت نبود

[1] Column family
[2] HBase
[3] Hypertable
[4] Cassandra
[5] Super column
[6] Null

مقدار واقعی، مقدار خالی نول ذخیره شوند در صورتیکه در پایگاه‌دادهای مبتنی بر ستون فقط مقادیر موجود کلید (صفت) مقدار ذخیره می‌شود(Hecht, R., S. Jablonski, 2011).

۲-۳-۱-۳ پایگاه داده مبتنی بر گراف

انواع پایگاه‌های ذکر شده تا کنون برای ذخیره سازی توزیع شده حجم زیادی از داده‌های نرمال نشده تعریف شده‌اند و برقراری ارتباط بین داده‌ها باید در سطح برنامه کاربردی صورت گیرد. بر خلاف این پایگاه‌ها پایگاه‌های مبتنی بر گراف برای ذخیره‌سازی ارتباطات بین داده‌ها ایجاد شده‌اند و از گره‌ها، ارتباطات بین آن‌ها و ویژگی‌های هر گره ساخته شده‌اند. بعبارتی دیگر، در این مدل پایگاه‌داده، داده‌ها و ارتباط بین آن‌ها بصورت شبکه‌ای از راس ها و یال ها مدل می شود و فرآیند مدل سازی و مشخص کردن روابط بین داده‌ها بسیار ساده است. به عنوان مثال برای پیدا کردن دوستان یک فرد در شبکه‌های اجتماعی از الگوریتم‌های کوتاهترین مسیر در گراف برای یافتن نتایج استفاده می‌شود در صورتی که در پایگاه‌داده‌های رابطه‌ای هزینه بیشتری صرف پیدا کردن چنین رابطه‌هایی می‌شود(Hecht, R., S. Jablonski, 2011). از میان ساختار های موجود پایگاه های داده غیر رابطه ای مدل‌های مبتنی بر گراف بیشترین کارایی را در داده‌هایی مانند گراف‌های شبکه‌های اجتماعی دارند. مثال هایی از این نوع پایگاه های داده نئوفورجی[1] و زرافه (جراف)[2] است.

۲-۳-۱-۴ پایگاه داده مبتنی بر سند

برخلاف پایگاه‌های داده رابطه‌ای که در آن‌ها داده‌ها در سطر و ستون ذخیره می شود، این مدل از پایگاه‌های داده غیررابطه‌ای داده‌ها را در سندهایی با فرمت هایی همچونJSON ، BSONو XML

[1] Neo4j
[2] Giraph

ذخیره می کنند. این مدل برای ذخیره داده‌های بدون ساختار بسیار مناسب است چرا که هر سند می تواند از تعداد متفاوتی فیلد[1] تشکیل شده باشد و لزومی ندارد که سند های مشابهی که در یک دسته بندی قرار می گیرند حتما از ساختار مشابهی پیروی کنند.

از طرفی پایگاه داده سند گرا، قابلیتهای بیشتری نسبت به پایگاه داده کلید-مقدار فراهم میکند. واحد ذخیره در این پایگاه یک سند است . سند یک شی است که مجموعه دلخواهی از ویژگیها را شامل می‌شود که به‌طور مثال میتواند در قالب[2] جی-سون ارائه شود. هر سند یک کلید منحصر به فرد دارد که توسط آن شناسایی می‌شود. ذخیره در قالب سند از جستجو بر اساس فیلدهای یک سند و شاخص گذاری پشتیبانی میکند. به‌طور معمول شمای داده از پیش تعریف شده ندارند. ذخیره سازی سند جدید با صفات جدید براحتی قابل انجام است. یک امتیاز این سیستمها نسبت به سیستمهای کلید-مقدار، پشتیبانی از قابلیت مجموعه‌ای از پرس‌وجوها روی یک سند با ایجاد چندین محدودیت روی فیلدهاست. این سیستم‌ها می‌توانند پرس‌وجوهای تجمعی را اجرا کنند، نتایج را مرتب کنند و از ایندکس گذاری روی فیلدهای یک سند پشتیبانی کنند(Hecht, R., S. Jablonski, 2011). نمونه‌هایی از این مدل پایگاه‌های داده غیر رابطه ای مبتنی بر سند(Hoberman, S. 2014) MongoDB، Elastic search (Divya, M. S., Goyal, S. K. 2013)می‌باشند.

۲-۳-۲ مروری بر مطالعات پیشین

امروزه پایگاه‌داده‌های NoSQL در شرکت‌های بزرگ دنیا جهت نگهداری و پردازش داده‌ها به طور گسترده مورد استفاده قرار می‌گیرند. البته به این معنی نیست که در این شرکت‌ها از پایگاه-داده‌های رابطه‌ای استفاده نمی‌شود بلکه NoSQLها برای کاربردهای خاص جستجو و پردازش و

[1] Field
[2] Jason

ذخیره‌سازی در کنار پایگاه‌داده‌های رابطه‌ای قرار می‌گیرند. به عنوان مثال پایگاه‌داده MongoDB در شرکت‌های تجاری مانند اتو[۱]، ای بی[۲]، گپ[۳]، ساکس فیفت اونیو[۴]،گیلت گروپ[۵] و ... . مورد استفاده قرار گرفته است (MongoDB, 2019) و یا پایگاه‌داده کاساندرا در شرکت‌های تجاری مانند کانستانت کانتکت[۶]، نت فلیکس[۷]، اینستاگرام[۸]، هولو[۹]، گو ددی[۱۰]، گیت هاب[۱۱]، کام کست[۱۲]، سرن[۱۳]، د ودر چنل[۱۴]، ردیت، اینتوییت[۱۵]، و .... استفاده می‌شود(Han, J., et al., 2011) . شایان ذکر است که شبکه اجتماعی تلگرام برای جستجو در هشتگ‌های خود از پایگاه‌داده کاساندرا استفاده می‌کند. این مثال‌ها نمونه‌هایی از شرکت‌های بزرگی است که پایگاه‌داده‌های NoSQL را بخدمت گرفته‌اند.

در ادامه به برخی از تحقیقاتی که به بررسی چالش‌ها و کاربردهای کلان‌داده‌ها و پردازش آن‌ها پرداخته‌اند اشاره می‌کنیم.

کوپر[۱۶] و همکارانش (۲۰۱۰) با ارائه چارچوب وای-سی-اس-بی[۱۷] (معیار ارزیابی سرویس ابری یاهو!) راهکاری را برای ارزیابی سرویس تحت ابر ارائه دادند. آن‌ها توسط چارچوب پیشنهادی خود داده‌های

[1] Otto
[2] eBay
[3] Gap
[4] Saks Fifth Avenue
[5] Gilt Group
[6] Constant contact
[7] Netflix
[8] Instagram
[9] Hulu
[10] GoDaddy
[11] GitHub
[12] Comcast
[13] Cern
[14] The Weather Channel
[15] Intuit
[16] Cooper
[17] YCSB (Yahoo! Cloud Service Benchmark)

تولید کرده و پایگاه‌داده‌های اچ-بیس[۱]، کاساندرا[۲]، پی-نات یاهو[۳]، مای اس-کسو-ال ساده شارد[۴] را مقایسه نمودند.

غزال و همکارانش (۲۰۱۳) سیستمی بنام بیگ-بنچ [۵]را برای تولید معیارسنجی[۶] های داده‌های کلان ارائه نمودند که از نظر تنوع و حجم داده، ساختار داده (ساخت‌یافته ، نیمه ه ساخت یافته و بدون ساختار) ویژگی‌های کلان داده‌ها را شبیه‌سازی کند.

آبراموا و همکارانش (۲۰۱۴) با ایجاد انواع مختلفی از بارکاری ها، پنج پایگاه‌داده غیر رابطه‌ای اچ-بیس، اورینتد دی-بی، ردیس، مانگو دی- بی[۷] را از نظر سرعت خواندن و نوشتن و به روزرسانی با یکدیگر مقایسه کرده‌اند. ردیس و کاساندرا در تمام آزمایشات نسبت به سایر پایگاه‌داده‌ها کارایی بهتری داشتند.

ماوریدیس[۸] و همکارانش (۲۰۱۷) از ابزارهای پردازش کلان‌داده برای آنالیز فایل لاگ وب سرور[۹] استفاده نمودند. آن‌ها با راه‌اندازی یک کلاستر ۶ گره‌ای کارایی امکانات هادوپ (Hadoop, 2017) و اسپارک (Chambers, B., & Zaharia, M. 2018)ذخیره‌سازی و پردازش کلان‌داده را با هم مقایسه کردند.

دده[۱۰] و همکارانش (۲۰۱۳) به ارزیابی الگوریتم مپ ردیوس [۱۱] در هادوپ و MongoDB پرداختند. آن‌ها برای ارزیابی خود از مجموعه داده ۳۰۰ گیگا بایتی سنسوس یو-اس [۱۲]استفاده کردند. در تمام

---

[1] Hbase
[2] Cassandra
[3] Yahoo!'s PNUTS
[4] Simple sharded MySQL
[5] BigBench
[6] Benchmark
7 MangoDB
[8] Mavridis
9 Web server log file Analysis
10 Dede
11 MapReduce
[12] U.S. Census dataset

بارهای کاری تست شده هادوپ کارایی بهتری از خود نشان داده است.

یکی از کاربردهای سیستم‌های کلان داده تجزیه و تحلیل موقعیت حرکتی افراد و تجزیه و تحلیل رفتار فرد یا بیمار برای حفظ سلامت و نظارت بر نحوه درست حرکت آن‌هاست. دولو و همکارانش (1395)با ارائه یک نرم‌افزار موبایل موقعیت مکان افراد را استخراج و از ابزارهای کلان‌داده برای پردازش آن استفاده نمودند. آن‌ها از محیط هادوپ[1] و ابزار هایو[2] و ابزار هایو برای استخراج الگوهای حرکتی در داده‌ها و کمک به سلامت افراد بهره بردند.

نوریان و همکارانش (۱۳۹۲) از تفسیر کلان-داده‌ها برای اعتبارسنجی معیارهای تجربی مربوط به توسعه مبتنی بر حمل ونقل عمومی و برنامه‌ریزی حمل و نقل شهری در خصوص سفرسازی استفاده نمودند. بدین منظور الگوی ترافیکی کاربری آموزش‌عالی با روش ارزیابی همبستگی تحلیل می‌گردد. با به کارگیری نگرش شهرسازی در پالایش کلان-داده‌های ترافیکی موجود و در دسترس برای دانشگاه ایالتی کالیفرنیا در شهر لوس‌آنجلس، الگوی ترافیکی سفرهای تحصیلی استخراج شده است.

نصری فلاح (۱۳۹۵) به بررسی کاربردها و چالش‌های کلان‌داده‌ها پرداخته است. مدبرفرزام و همکارانش (۱۳۹۴) ضمن بررسی چالش‌های کلان‌داده‌ها ویژگی‌ها و انواع پایگاه‌داده‌های NoSQL را بررسی کرده‌اند.

در نهایت با بررسی پژوهشهای پیشین در این حوزه و در نظرگرفتن نوع داده های در دست در مرکز منطقه ای و هدف این پژوهش که بررسی عملیات جستجو و بازیابی تمام متن بر روی آن‌هاست؛ پایگاه‌های داده غیررابطه ای /سیستمهای NOSQL محبوب و منتخب MongoDB و

[1] Hadoop
[2] Hive

Elasticsearch که در دسته مبتنی بر سند قرار می گیرند برای ارزیابی درقیاس با پایگاههای داده رابطه ای لحاظ شده اند؛ بنابراین در ادامه به شرح مختصری در خصوص آنها می پردازیم.

۲-۳-۳ پایگاه دادهMongoDB

MongoDB یک پایگاه داده غیر رابطه ای مبتنی بر سند است که رکورد های داده را تحت قالب BSONذخیره می کند. مفاهیم جدول[۱]، ردیف[۲] و ستون[۳]که در پایگاه های داده رابطه ای مطرح هستند در این پایگاه داده به ترتیب با مفاهیم جدیدی به نام های; کلکسیون[۴]، سند[۵] و فیلد[۶] جایگزین می شوند. همچنین برخلاف پایگاه‌های داده رابطه ای که برای بازیابی اطلاعات مرتبط که در جدول های متفاوتی قرار گرفته اند، احتیاج به SQL Join می‌باشد، در MongoDB به این علت که اطلاعات مرتبط به هم بصورت سندهای تعبیه شده[۷] در یک جا قرار می گیرند، مفهوم Join وجود ندارد. راهکار این پایگاه داده برای ارائه قابلیت مقیاس پذیری افقی، خرد کردن به روش شاردینگ [۸] است. در صورتی که این قابلیت بر روی یک پایگاه داده فعال شود، کلیکسیونهای آن پایگاه داده می توانند بین نمونه های[۹] متفاوتی از MongoDB که در یک[۱۰] خوشه خرد شده قرار دارند، توزیع شوند. در صورت فعال سازی قابلیت شاردینگ بر روی یک کلکسیون ، سندهای آن کلکسیون بر روی نمونه

[1] Table
[2] Row
[3] Column
[4] Collection
[5] Document
[6] Field
[7] Embeded Document
[8] Sharding
[9] Instances
[10] Sharded Cluster

های مجزا توزیع خواهند شد. وظیفه توزیع متعادل این سندها بر روی نمونه های مختلف، بر عهده بالانس کننده[۱] می‌باشد.(ElasticSearch, 2019; Dixit, B.,2016; MongoDb, 2019)

۲-۳-۴ پایگاه داده Elasticsearch

Elasticsearch بیش از آنکه به عنوان یک پایگاه‌داده شناخته شود، به عنوان یک موتور جستجوی- تمام متن [۲] و توزیع شده مطرح است که بر پایه کتابخانه لوسنس [۳]و به زبان جاوا [۴]توسعه داده شده است. پس از انتشار نسخه اولیه Elasticsearch در سال ۲۰۱۰، این موتور جستجو به طور گسترده در سازمان های بزرگی همچون ناسا[۵]، ویکی پدیا[۶]و گیت هاب[۷] N به کار گرفته شده است. آخرین نسخه های انتشاریافته از Elasticsearch ، به هدف اینکه علاوه بر موتور جستجو بودن به عنوان پایگاه داده نیز مورد اعتماد کاربران قرار گیرد، بیشتر بر ویژگی انعطاف پذیری [۸]متمرکز بوده است.

Elasticsearch بر پایه معماری رست[۹] بنا شده و برقراری ارتباط با این پایگاه داده برای انجام عملیات مختلف و تغییر تنظیمات، توسط درخواست های HTTP صورت می گیرد. در این پایگاه داده، اصطلاحات کلکسیون و پایگاه داده که در MongoDB معرفی شدند، به ترتیب با مفاهیممپینگ[۱۰] و ایندکس[۱۱] ایندکس جایگزین می شوند. سند ها نیز در Elasticsearch با فرمت جی-سون[۱۲] ذخیره می شوند. Elasticsearch نیز به منظور فراهم کردن قابلیت مقیاس پذیری افقی مفهوم شاردینگ و

[1] Balancer
[2] Full-Text Search
[3] Lucene
[4] Java
[5] Nasa
[6] WikiPedia
[7] Github
[8] Resiliency
[9] REST
[10] Mapping
[11] Index
[12] JSON

خرد کردن را معرفی می کند. در یک Elasticsearch کلاستر این قابلیت وجود دارد که ایندکس ایندکس های متفاوت با تعداد شارد خرد شده متفاوتی تعریف شوند. به عبارتی کاربر این اختیار را دارد که هر ایندکس ایندکس را به تعداد قطعات متفاوتی بشکند و بر روی نود های کلاستر توزیع کند(Chambers, B., & Zaharia, M. 2018) .

۲-۳-۵ مقایسه ویژگی‌های پایگاه‌های داده MongoDB و Elasticsearch

با توجه به اینکه پایگاه‌های داده Elasticsearch و MongoDB هر دو جزء دسته بندی پایگاه‌های داده غیر رابطه ای مبتنی بر سند قرار می گیرند، نقاط مشترک متعددی بین آن ها وجود دارد. از جمله دارا بودن ویژگی هایی همچون پذیرفتن داده‌های بدون ساختاربودن [1]، ذخیره و بازیابی داده ها بصورت توزیع شده [2](و دسترسی بالا[3] قابل ذکر است .

با این وجود تفاوت های اساسی بین آن‌ها وجود دارد که باعث شده هر کدام در جایگاه خاص خود مورد استفاده قرار گیرد. از جمله این تفاوت ها می توان به موارد زیر اشاره کرد (MongoDb Manual Document, 2019; Github, 2019; Elasticsearch Resiliency Status, 2019)

- MongoDB به عنوان یک پایگاه داده همه منظوره [4] شناخته می شود و در نقطه مقابل Elasticsearchبرای هدف خاص تری طراحی شده است و آن قابلیت ایندکسینگ و جستجوی قدرتمند و توزیع شده بین سند ها در ابعاد بزرگ است.
- با توجه به همین تفاوت ماهیت این دو پایگاه داده دور از انتظار نیست که Elasticsearch به اندازه MongoDB ، به عنوان یک ابزار ذخیره سازی داده‌ها، انتظارات را برآورده نکند. همینطور قابلیت

[1] Scheme Free
[2] Sharding
[3] High Availability
[4] Genral Purpose

های ایندکسینگ و جستجوی تمام متن قدرتمند Elasticsearch را نمی توان از MongoDB انتظار داشت.

- علی رغم تلاش های انجام شده، بزرگترین ضعف Elasticsearch همچنان عدم قابلیت اعتماد[1]بودن آن در نگهداری و حفظ داده‌ها به عنوان یک سند است. این مشکل معمولا زمانی نمود پیدا می کند که کلاستر تغییر وضعیت داده باشد و نتیجتا باعث می شود که Elasticsearch به عنوان پایگاه داده اصلی مورد اطمینان نباشد.
- Elasticsearch از واسط ای-پی- آی رست[2] و پروتکل قدرتمند HTTP بهره می گیرد و این مورد از نقاط قوت آن نسبت به MongoDB به شمار می آید.

در پایان این فصل به مقایسه ای بین پایگاههای داده رابطه ای و غیر رابطه ای کاربردی در پژوهش حاضر در قالب جدول های ۲-۳ و ۲-۲ می پردازیم.

جدول ۲-۲ مقایسه ویژگیهای کلی پایگاههای داده رابطه ای و غیر رابطه ای مورد بررسی پژوهش حاضر

مدل داده ها	کارآیی	مقیاس پذیری	قابلیت انعطاف پذیری	پیچیدگی	عملکرد
پایگاه داده رابطه ای	متغیر	متغیر	پایین	متوسط	متغیر
پایگاه داده غیر رابطه ای	بالا	بالا	بالا	متغیر (پایین)	متغیر

[1] Reliable
[2] API REST

جدول ۲-۳ مقایسه مزایا و معایب پایگاههای داده رابطه ای و غیر رابطه ای کاربردی در پژوهش حاضر

(Chen, J. K. et al., 2019; DB-Engines Ranking, 2017; Oracle-RDBMS, 2017; Kenler, E., 2015; Li, Y, et al., 2013)

موارد مقایسه	پایگاه داده رابطه ای /Relational Data Base		پایگاه داده غیررابطه Non-Relational Data Base (NoSQL) ای	
	Microsoft SQL Server	Maria DB	MangoDB	Elastic Search
شناسنامه	• توسعه یافته : Microsoft Corporation • انتشار اولیه : ۲۴ آپریل ۱۹۸۹ • سیستم عامل:Linux, Microsoft Windows Server, Microsoft Windows • زبان توسعه :C, C++	• توسعه یافته : MariaDB Corporation AB, MariaDB Foundation • انتشار اولیه : ۲۹ اکتبر ۲۰۰۹ • سیستم عامل: Windows , Solaris , FreeBSDLinux, X 10.7 OS , Vista • زبان توسعه:Windows, macOS	• توسعه یافته : MongoDB Inc. • انتشار اولیه : ۱۱ فوریه ۲۰۰۹ • سیستم عامل: Solaris ,Windows و FreeBSD , Vista , Linux • زبان توسعه : جاوااسکریپت ، پایتون ، C++, Go,	• توسعه یافته : Elastic NV • انتشار اولیه : ۸ فوریه ۲۰۱۰ • سیستم عامل: Windows Vista and later, Linux, OS X 10.7 and later, Solaris, FreeBSD • زبان توسعه : جاوا

کاربرد (use case)				
• اگر انتظار دارید داده هایتان در آینده تغییر کند یا ساختار داده ها در آینده مشخص نیست. • اگر برنامه ی شما درخواست های دریافت خوانش (Read) زیادی دارد اما داده ها را زیاد تغییر نمی دهید. • اگر بعدا به مقیاس دهی افقی نیاز پیدا خواهید کرد باید Elastic Search -NoSQLرا انتخاب کنید.	• اگر انتظار دارید داده هایتان در آینده تغییر کند یا ساختار داده ها در آینده مشخص نیست . • اگر برنامه ی شما درخواست های دریافت خوانش (Read) زیادی دارد اما داده ها را زیاد تغییر نمی دهید. • اگر بعدا به مقیاس دهی افقی نیاز پیدا خواهید کرد باید MongoDB -NoSQLرا انتخاب کنید.	• اگر انواع داده های مختلفی دارید که در قسمت های مختلف برنامه تان از آن ها استفاده می کنید بهتر است از SQL ها استفاده کنید . • اگر داشتن یک ساختار ثابت برایتان مهم است و احتمال تغییر داده هایتان وجود ندارد می توانید از SQL ها استفاده کنید.	• اگر انواع داده های مختلفی دارید که در قسمت های مختلف برنامه تان از آن ها استفاده می کنید بهتر است از SQL ها استفاده کنید. • اگر داشتن یک ساختار ثابت برایتان مهم است و احتمال تغییر داده هایتان وجود ندارد می توانید از SQLها استفاده کنید.	

• بدلیل ساختار اولیه طراحی با ریسک کمتری از بابت حملات پایگاه داده ای مواجه است • انعطاف پذیری بیش از حد ممکن است باعث تنبل شدن نیروی سازمانی و پیاده نکردن ساختاری مشخص برای پایگاه داده تان شود.	• بدلیل ساختار اولیه طراحی با ریسک کمتری از بابت حملات پایگاه داده ای مواجه است • انعطاف پذیری بیش از حد ممکن است باعث تنبل شدن نیروی سازمانی و پیاده نکردن ساختاری مشخص برای پایگاه داده تان شود.	• ریسک بالای در معرض قرار گرفتن حملات SQL-INJECTION • عدم مقیاس پذیری افقی و درنتیجه ساپورت حجم محدودی داده	• ریسک بالای در معرض قرار گرفتن حملات SQL-INJECTION • عدم مقیاس پذیری افقی و درنتیجه ساپورت حجم محدودی داده	**ریسک (Risk)**

• بهره بردن از واسط API REST و پروتکل قدرتمند HTTP • قابلیت های ایندکسینگ و جستجوی تمام متن قدرتمند • ذخیره و بازیابی داده‌های بدون ساختار در قالب فرمت های مختلف (Schema Free) • ذخیره و بازیابی داده ها بصورت توزیع شده • انعطاف پذیری و سازگاری بیشتر • دسترسی بالا(High Availability) • عدم نیاز به استفاده از پرس و جوهای آزمایشی پیچیده • افزایش سرعت توسعه برنامه‌های مختلف • افزایش سرعت دسترسی به داده‌های مختلف	• یک پایگاه داده همه منظوره (General Purpose) با خصوصیات بهینه برای داده های بدون ساختار ذخیره و بازیابی داده‌های بدون ساختار در قالب فرمت های مختلف (Schema Free) • ذخیره و بازیابی داده ها بصورت توزیع شده • انعطاف پذیری و سازگاری بیشتر • دسترسی بالا (High Availability) • عدم نیاز به استفاده از پرس و جوهای آزمایشی پیچیده	• از محبوب ترین و کاربرپسندترین پایگاه‌های داده رابطه‌ای است که نسبت به MySQL برتریهایی دارد. • قدرت اصلی این مدل، در استفاده از مفهوم جدول است که روشی بصری، کارآمد و تا حدودی انعطاف پذیر برای ذخیره و دسترسی به اطلاعات ساختاریافته است. • مهاجرت راحت از سایر سیستم های پایگاه داده به MariaDB از دیگر مزایای آن است. • منبع باز و رایگان	• قدرت اصلی این مدل، در استفاده از مفهوم جدول است که روشی بصری، کارآمد و تا حدودی انعطاف پذیر برای ذخیره و دسترسی به اطلاعات ساختاریافته است. • از محبوب ترین و کاربرپسندترین پایگاه‌های داده رابطه‌ای است.	**مزایای کلی (Pros)**

• قابلیت مقیاس پذیری عمودی و افقی بدون حد نصاب برای داده های عظیم • قابلیت مقیاس پذیری افقی در قالب یک سیستم توزیع شده • منبع باز و رایگان	• افزایش سرعت توسعه برنامه‌های مختلف • افزایش سرعت دسترسی به داده‌های مختلف • قابلیت مقیاس پذیری عمودی و افقی بدون حد نصاب برای داده های عظیم • قابلیت مقیاس پذیری افقی در قالب یک سیستم توزیع شده • منبع باز و رایگان			

• بزرگترین ضعف Elasticsearch عدم reliable بودن آن در نگهداری و حفظ داده‌ها به عنوان یکپایگاه داده است. • انعطاف پذیری بیش از حد ممکن است باعث پیاده نکردن ساختاری مشخص و مدون برای پایگاه داده سازمانی شود • داده های تکراری باعث می شوند که برای انجام عملیاتی خاص مانند به روز رسانی مجبور شوید چندین دستور به روز رسانی را برای چندین مقدار مختلف اجرا کنید.	• ضعف نسبی در ابزار ایندکسینگ و جستجوی تمام متن داده‌ها • انعطاف پذیری بیش از حد ممکن است باعث پیاده نکردن ساختاری مشخص و مدون برای پایگاه داده سازمانی شود. • داده های تکراری باعث می شوند که برای انجام عملیاتی خاص مانند به روز رسانی مجبور شوید چندین دستور به روز رسانی را برای چندین مقدار مختلف اجرا کنید	بطور کلی محدودیتهایی از نظر مقیاس پذیری، موازی سازی عملیات و هزینه به شرح زیر دارد: • بازیابی اطلاعات پیچیده که منجر به افت عملکرد عملیات می‌شود. • انعطاف پذیری کم به دلیل ساختار از پیش تعیین شده. • بسیار سخت و یا غیر ممکن بودن تغییر ساختار جداول در آینده • روابط ممکن است باعث ایجاد پرس و جوهای آزمایشی بسیار پیچیده ی شوند.	بطور کلی محدودیتهایی از نظر مقیاس پذیری، موازی سازی عملیات و هزینه به شرح زیر دارد: • بازیابی اطلاعات پیچیده که منجر به افت عملکرد عملیات می‌شود. • انعطاف پذیری کم به دلیل ساختار از پیش تعیین شده. • بسیار سخت و یا غیر ممکن بودن تغییر ساختار جداول در آینده • روابط ممکن است باعث ایجاد پرس و جوهای آزمایشی بسیار پیچیده ی شوند.	**معایب کلی (Cons)**

		• مقیاس دهی افقی کار سختی است و در اکثر مواقع مقیاس دهی فقط عمودی است بنابراین اگر شرکت شما بسیار بزرگ شود به حد نصاب خاصی خواهید رسید. • هیچگونه استاندارد زبانی پرس و جوهای آزمایشی ندارد.	• مقیاس دهی افقی در کار سختی است و در اکثر مواقع مقیاس دهی فقط عمودی است بنابراین اگر شرکت شما بسیار بزرگ شود به حد نصاب خاصی خواهید رسید • هزینه لیسانس بالا.	

• Elasticsearch طراحی شده است برای اهداف خاص از جمله قابلیت ایندکسینگ و جستجوی قدرتمند و توزیع شده بین Document ها در ابعاد بزرگ • Elasticsearchبه اندازه MongoDB، به عنوان یک ابزار ذخیره سازی داده‌ها، انتظارات را برآورده نمی کند • Documentها نیز در Elasticsearchبا فرمت JSON ذخیره می شوند. • به هدف اینکه علاوه بر موتور جستجو بودن به عنوان پایگاه داده نیز مورد اعتماد کاربران قرار گیرد، بیشتر بر ویژگی resiliency متمرکز بوده است.	• MongoDB یک پایگاه داده همه منظوره (General Purpose)با خصوصیات بهینه برای داده های بدون ساختار است. • به اندازه Elasticsearch ، به عنوان یک ابزار قدرتمند ایندکسینگ و جستجوی تمام متن داده‌ها، انتظارات را برآورده نمی کند	• پس از انتقال مالکیت کامل پروژه MySQL از شرکت MySQL ABبه شرکت Oracleدر بین سال های ۲۰۰۸ و ۲۰۰۹، توسعه دهندگان اصلیMySQL ، پایگاه داده MariaDB را به عنوان یک fork از MySQL معرفی کردند و تا به امروز این پایگاه‌داده نوپا در بسیاری از شرکت ها جایگزین MySQLشده است • MariaDB در مقایسه با MySQL از موتورهای بیشتری پشتیبانی می کند. • MariaDB با توجه به منبع باز بودن در مقایسه با	• از قدیمیترین و شناخته شده ترین پایگاه داده های رابطه ای با محبوبیت و قابلیت کاربری خوب با استفاده از رابط کاربری آسان • در این نوع پایگاه داده هر سطر یک جدول، یک رکورد با یک کلید شناسایی منحصر به فرد است و هر ستون نمایانگر یک ویژگی در مورد رکورد های آن جدول است • پایگاه داده ای است که داده‌های مرتبط به هم را ذخیره کرده و دسترسی به آن ها را فراهم می آورد. به	**سایر نکات**

• Elasticsearch نیز به منظور فراهم کردن قابلیت Horizontal Scalabilityمفهوم شاردینگ را معرفی می کند		MySQL به طور مداوم در حال توسعه است. • MariaDB همچنین یک پایگاه داده خوشه ای برای استفاده تجاری ارائه می دهد ، که همچنین امکان تکثیر چند اسناد را فراهم می کند. • MariaDB برای کارآیی بالا بهینه شده است و برای مقادیر زیادی مجموعه داده بسیار قدرتمندتر از MySQL است.	عبارتی این نوع پایگاه‌های داده مبتنی بر مدل Relationalمی باشند که یک راه شهودی و قابل درک برای نشان دادن اطلاعات در جداول است. • در این نوع پایگاه داده هر سطر یک جدول، یک رکورد با یک کلید شناسایی منحصر به فرد است و هر ستون نمایانگر یک ویژگی در مورد رکورد های آن جدول است	

فصل سوم: روش شناسی پژوهش

(مدل‌سازی و پیاده‌سازی آزمایشگاهی)

۳–۱ مقدمه

با در نظر داشتن هدف اصلی پژوهش حاضر مبنی بر بررسی عملکرد جستجوی تمام متن بر پایگاه‌های داده رابطه‌ای در قیاس با پایگاه داده های غیر رابطه ای مبتنی بر متن و متناسب با پژوهشهای انجام شده در خصوص پایگاههای داده غیر رابطه ای (MongoDB, 2019; Cooper B.f. et al., 2010) ، دو مدل پایگاه‌های داده غیر رابطه‌ای MongoDB و Elasticsearch به عنوان سیستمهای NOSQL محبوب جهت مدل سازی و پیاده سازی آزمایشگاهی برگزیده شدند. به روش مشابه Microsoft SQL Serverو MariaDB نیز بعنوان پایگاه داده های رابطه ای منتخب برگزیده شدند. سپس سایر مراحل پیاده سازی و مدل سازی سخت افزاری و نرم افزاری مورد نیاز در حالت آزمایشگاهی(مشروح در ادامه فصل) صورت گرفت تا بتوان به ارزیابی مد نظر در دو فاز زیر بپردازیم :

- بررسی عملکرد بارهای کاری ۱۰۰٪ خوانش و ۱۰۰٪ اسکن برای پایگاه‌های داده مورد بحث به کمک ارزیابی وای-سی-اس-بی[1]

[1] YCSB Benchmark

- بررسی عملکرد پرس و جوهای آزمایشی جستجوی تمام متن برای پایگاه‌های داده مورد بحث و همچنین در نظر گرفتن تاثیر قابلیتl مقیاس پذیری افقی بر عملکرد این پرس و جوهای آزمایشی در پایگاه‌های داده غیر رابطه‌ای MongoDB و Elasticsearch

۳-۲ مشخصات فنی سیستم و نسخه پایگاه‌های داده مورد استفاده

۳-۲-۱ مشخصات فنی سیستم

برای انجام ارزیابی‌ها از هفت سیستم با مشخصات فنی مندرج درجدول زیر استفاده شده است.

جدول ۳-۱ مشخصات سیستمها

نام سیستم	IPآدرس
سیستم ۱	172.16.13.26
سیستم ۲	172.16.13.25
سیستم ۳	172.16.13.24
سیستم ۴	172.16.13.23
سیستم ۵	172.16.13.22
سیستم ۶	172.16.13.21
سیستم ۷	172.16.13.20

مشخصات سخت افزاری سیستم ۱ به شرح زیر می باشد:

- Operating System: Linux CentOS 7
- Architecture: x86 _64
- Processor: Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz
- Memory: 16GiB DIMM DDR4 Synchronous 2400 MHz (0.4 ns)
- HDD: 2TB WDC WD20EZRZ-00Z

در قسمت ارزیابی جستجو و بازیابی تمام متن، برای بررسی تاثیرشاردینگ یامقیاس پذیری افقی، ۶ سیستم دیگر (سیستم ۲ الی ۶) با ویژگی های مشترک سخت افزاری، برای میزبانی شارد ها استفاده شده است.

- Operating System: Linux CentOS 7
- Architecture: x86 _64
- Processor: Intel(R) Core(TM) i7 CPU 920 @ 2.67GHz
- Memory: 11GiB DIMM DRAM EDO
- HDD: 429GB Virtual disk

۳-۲-۲ نسخه پایگاه‌های داده مورد استفاده

نام پایگاه داده و نسخه های مورد استفاده هر کدام در این پژوهش در جدول زیر به اختصار آمده است.

جدول ۳-۲ پایگاههای داده مورد استفاده

نام پایگاه داده	نسخه
Microsoft SQL Server 2019 (CTP3.2)	15.0.1800.32
Elasticsearch	5.5.1
MariaDB Server	10.3.17
MongoDB	4.0.9

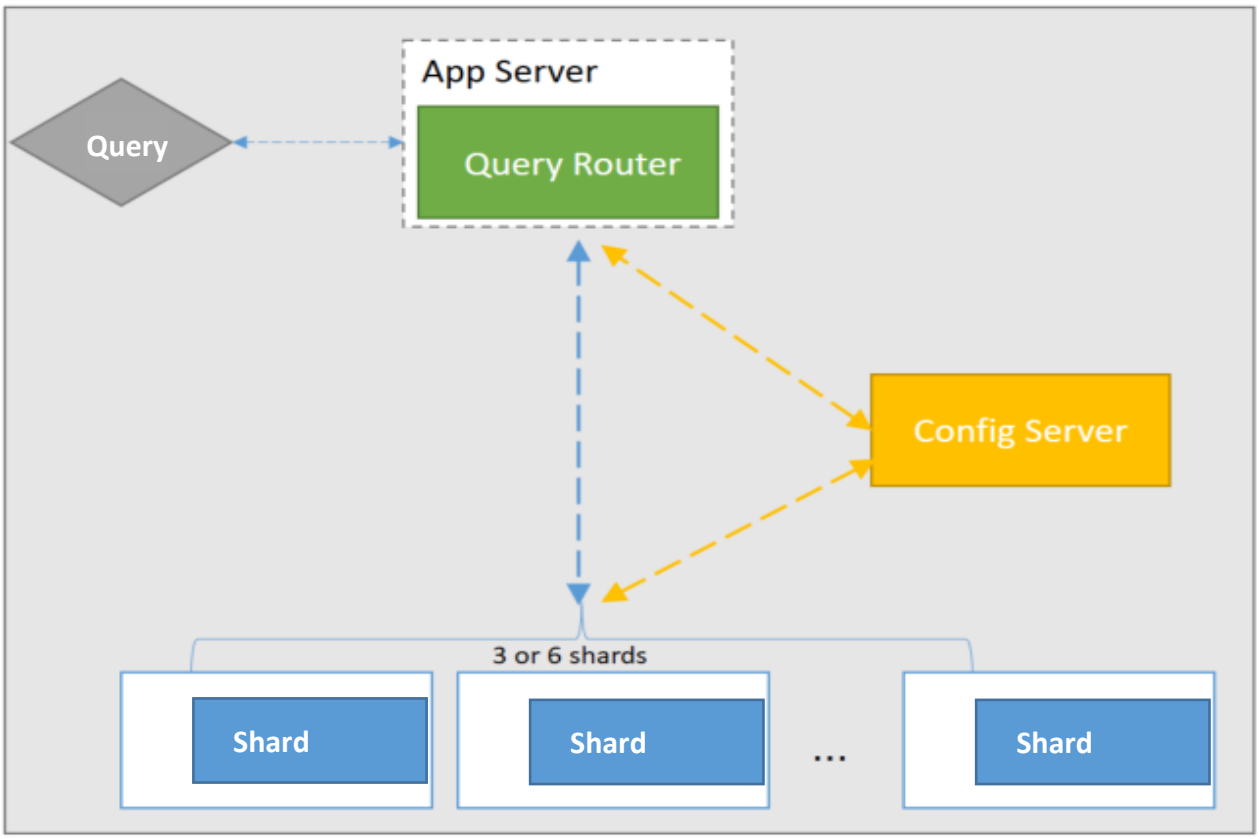
۳-۳ کلیات مدل سازی و پیاده سازی پایگاه‌های داده

در فاز نخست که در قسمت ۴-۲ به شرح جزئیات آن پرداخته می شود، چهار پایگاه داده Microsoft SQL Server، MariaDB، MongoDB و Elasticsearch به صورت تک نمونه [1]روی سیستم 1 (که مشخصات فنی آن در قسمت ۳-۳ آمده است) پیاده سازی شده و همچنین اجرای امعیار ارزیابی وای-سی-اس-بی[2] بر روی این پایگاه‌های داده انجام می شود.

در فاز بعدی که در قسمت ۴-۳به شرح جزئیات آن پرداخته می شود به بررسی مقایسه ای بین پایگاههای داده رابطه ای و غیر رابطه ای در جستجوی تمام متن در دو حالت بدون خوشه بندی (۴-۳-۱) و با خوشه بندی (۴-۳-۲) می پردازیم. در حالت خوشه بندی، کلاسترهای MongoDB و Elasticsearchنیز در حالت ۳شارد و ۶ شارد[3] پیاده سازی و عملکرد پرس و جوهای آزمایشی جستجوی تمام متن بر روی آن ها نیز بررسی می شود.

یک کلاستر در MongoDB و یا اصطلاحا کلاستر شارد و خرد شده[4] از سه جزء روتر پرسش وپاسخ [5](برای دریافت درخواست هایکلاینت)، کانفیگ سرور[6] و شارد/خرد شده[7] تشکیل شده است. در شکل۳-۱ معماری اجزای تشکیل دهنده MongoDB و ارتباط بین آنها نمایش داده شده است .

[1] Single Instance
[2] YCSB Benchmark
[3] Shard
[4] Sharded Cluster
[5] Query Router
[6] Config Server
[7] Shard



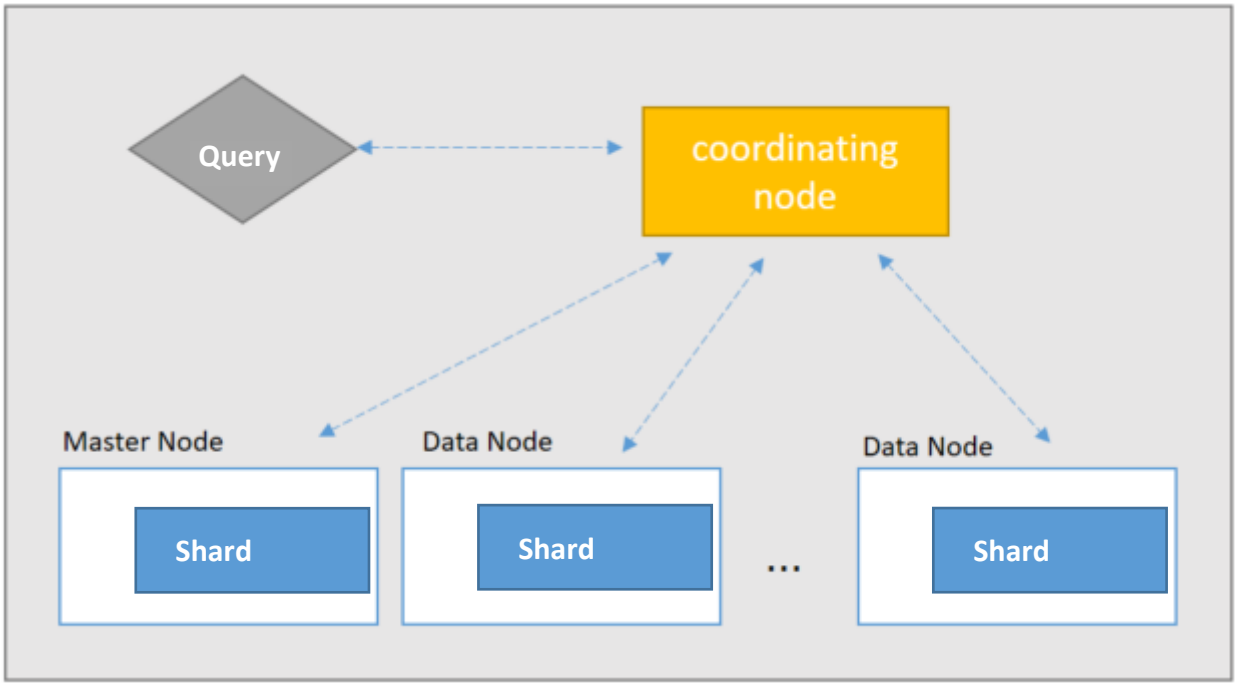
شکل ۳-۱ معماری پیاده سازی شده اجزای تشکیل دهنده بر بستر MongoDB

در این پژوهش برای پیاده سازی یک کلاستر شارد و خرد شده [1]از سیستم ۱ به عنوان میزبان بخش های سرور کانفیگ[2] و روتر پرسش و پاسخ[3] و از سایرسیستم ها برای میزبانی شاردها استفاده شده است (در حالت ۶ شارد از هر شش سیستم و در حالت ۳شارد فقط از سه سیستم استفاده شده است). جزئیات پیاده سازی یک کلاستر شارد خرد شده در (MongoDB Manual Document, 2019)که قسمتی از مستندات رسمی MongoDB است، آمده است.

[1] Sharded Cluster
[2] Config Server
[3] Query Server

یک کلاستر Elasticsearch مجموعه ای از نودهاست که هر کدام می توانند نقش هایی از قبیل مستر[1]، داده[2]، کوردیناتینگ[3] ویا ترکیبی از آن ها را داشته باشند (ElasticSearch, 2019). در این پژوهش، سیستم 1 به عنوان نود کوردیناتینگ [4]تنظیم شده که صرفا درخواست های کلاینت را دریافت کند. سیستم ۷ هم به عنوان نود داده [5](برای میزبانیشارد) و هم به عنوان مستر نود[6] (برای مدیریت کلاستر) استفاده شده و ۵ سیستم دیگر صرفا به عنوان نود داده تنظیم شده است. جزئیات پیکربندی نودها در (ElasticSearch Manual Document1, 2019 ; ElasticSearch Manual Document2, 2019)که قسمتی از مستندات رسمی Elasticsearch می‌باشد آمده است. در شکل ۳-۲ معماری اجزای تشکیل دهنده ElasticSearch و ارتباط بین آنها نمایش داده شده است.

[1] Master
[2] Data
[3] Coordinating
[4] Coordinating Node
[5] Node Data
[6] Master Node



شکل ۳-۲ معماری پیاده سازی شده اجزای تشکیل دهنده بر بستر ElasticSearch

فصل چهارم: یافته‌های پژوهش

۴-۱ مقدمه

در این فصل به بررسی و ارزیابی نتایج و یافته های حاصل از سناریوهای مختلفی که با هدف بررسی مقایسه ای عملکرد کلی و جستجوی تمام متن پایگاه های داده رابطه ای SQL Server و MariaDBو پایگاه های داده غیر رابطه ای MongoDB و Elasticsearch در دوبخش کاملا مجزا پرداخته می‌شود. در بخش اول آزمایشات، ابزار مورد استفاده معیار ارزیابی سرویس ابری یاهو![1] و یا به اختصار وای-سی- اس-بی[2] است که در بسیاری از تحقیقات آکادمیک در زمینه مقایسه پایگاه‌های داده مختلف به عنوان معیار ارزیابی[3] استاندارد مورد استفاده قرار می گیرد. در بخش دوم نیز به بررسی عملکرد جستجوی تمام متن برای پایگاه‌های داده SQL Server، MariaDB، MongoDB و Elasticsearchبا استفاده از چهار نوع پرس و جوهای آزمایشی[4] پرداخته خواهد شد. جزییات بیشتر نتایج این آزمایشات با انواع سناریوهای طراحی شده در قسمتهای ۴-۲ و ۴-۳ بتفصیل قابل مطالعه و ارزیابی است.

[1] Yahoo! Cloud Service Benchmark
[2] YCSB
[3] Benchmark
[4] Query

۴-۲ بررسی عملکرد پایگاههای داده رابطه ای و غیر رابطه ای در حالت اجرای YCSB Benchmark

در این قسمت به بررسی عملکرد کلی پایگاه های داده رابطه ای SQL Server و MariaDB و پایگاه های داده غیر رابطه ای MongoDB و Elasticsearch ومقایسه آنها با یکدیگر می پردازیم. ابزار مورد استفاده دراین قسمت معیار ارزیابی سرویس ابری یاهو![1] و یا به اختصار وای-سی-اس-بی است که در بسیاری از تحقیقات آکادمیک در زمینه مقایسه پایگاه‌های داده مختلف به عنوان معیار ارزیابی استاندارد مورد استفاده قرار می گیرد. هدف اصلی پروژهوای-سی-اس-بی ایجاد یک چارچوب و مجموعه ای از بارهای کاری متداول برای ارزیابی عملکرد سرویس های پایگاه داده مختلف است. این پروژه از دو قسمت اصلی تشکیل شده است (Yahoo! Cloud Serving Benchmark (YCSB), 2019):

- کلاینت وای-سی-اس-بی که بارهای کاری [2] قابل توسعه را تولید می کند.
- بارهای کاری هسته مرکزی[3] که مجموعه ای بارهای کاری اصلی هستند که توسط تولید کننده[4] اجرا می شوند.

YCSB که به زبان جاوا توسعه داده شده است برای پایگاه های داده متعددی اینترفیس ارائه کرده و همچنین با توجه به متن باز بودن این پروژه برای سایر پایگاه داده ها نیز قابل استفاده و توسعه توسط افراد مختلف است.

[1] Yahoo! Cloud Service Benchmark
[2] Workload
[3] Core Workloads
[4] Generator

در این پژوهش با توجه به اهداف پژوهش، دو بارکاری[1] مورد استفاده قرار گرفته است. بارکاری تحت عنوان WorkloadA که تماما شامل عملیات خوانش[2] می‌شود (معادل همان WorkloadC که بطور پیش فرض در YCSB تعریف شده است) و WorkloadB که تماما شامل عملیات اسکن[3] می‌باشد. اجرای هر بارکاری شامل دو مرحله است. در مرحله اول با مشخص کردن تعداد رکورد های مورد نظر، بایستی رکورد ها در پایگاه داده بارگذاری[4] شوند و در مرحله بعدی با مشخص کردن تعداد عملیات مورد نظر، بارکاری بر روی پایگاه داده اجرا[5] شود. همچنین ساختار هر رکورد بدین صورت است که هر کدام شامل ۱۰ فیلد می باشند و اندازه هر فیلد ۱۰۰ بایت است.

در این پژوهش در هر مرحله بارگذاری و اجرای بار کاری، تعداد رکورد ها و تعداد عملیات برابر در نظر گرفته شده و به ازای عدد های هزار، ده هزار و صد هزار نتایج جمع آوری شده است.

هنگام اجرای هر بارکاری به منظور بهره وری بیشتر از منابع سیستم می توان از قابلیت چند رشته ای [6]استفاده کرد و تعداد رشته های[7] مورد نظر را تعیین کرد. در این پژوهش به ازای تعداد رشته های ۱، ۲، ۴ و ۸ نتایج جمع آوری و مورد بحث واقع شده است. پارامتر های درنظر گرفته شده برای مقایسه توانش زمانی[8] و میانگین تاخیر زمانی[9] است. همچنین اجرای هر بارکاری با مشخصات خاص، برای ۵ مرتبه تکرار شده و نمودار ها با توجه به میانگین این نتایج رسم شده اند.

---

[1] Workload
[2] Read Operation
[3] Scan
[4] Load
[5] Run
[6] Multi-threading
[7] Thread

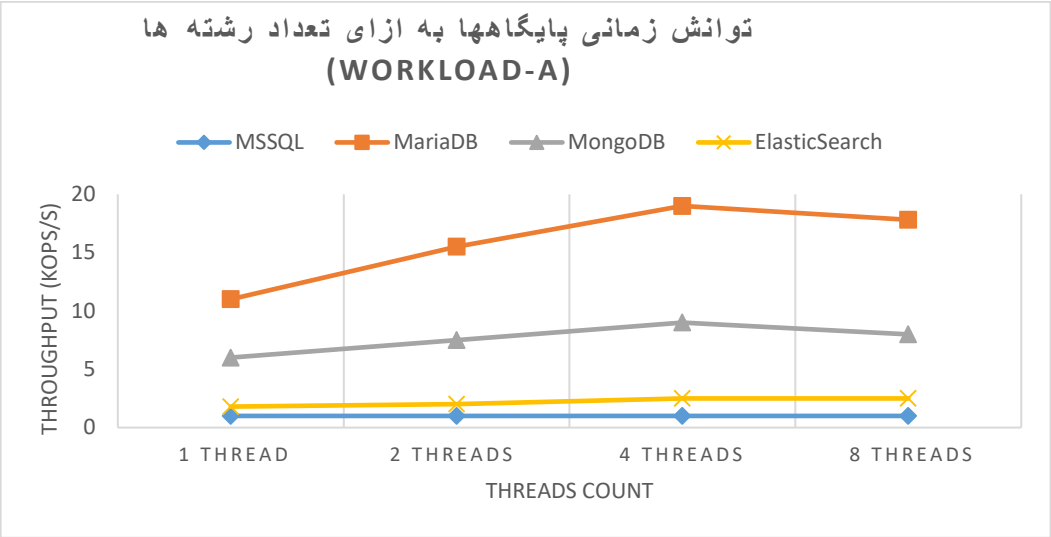
[8] Throughput :توانش زمانی یا تعداد عملیات انجام شده در واحد زمان
[9] Latency : در ترجمه لفظ latencyزمان بازیابی داده، زمان پاسخگویی به یک درخواست و تاخیر زمانی الفاظی است که در طول متن با توجه به ماهیت و بافت متن تخصصی با آن برخورد خواهیم داشت.

۴-۲-۱ بررسی عملکرد پایگاههای داده رابطه ای و غیر رابطه ای به ازای تعداد رشته های[1] متفاوت

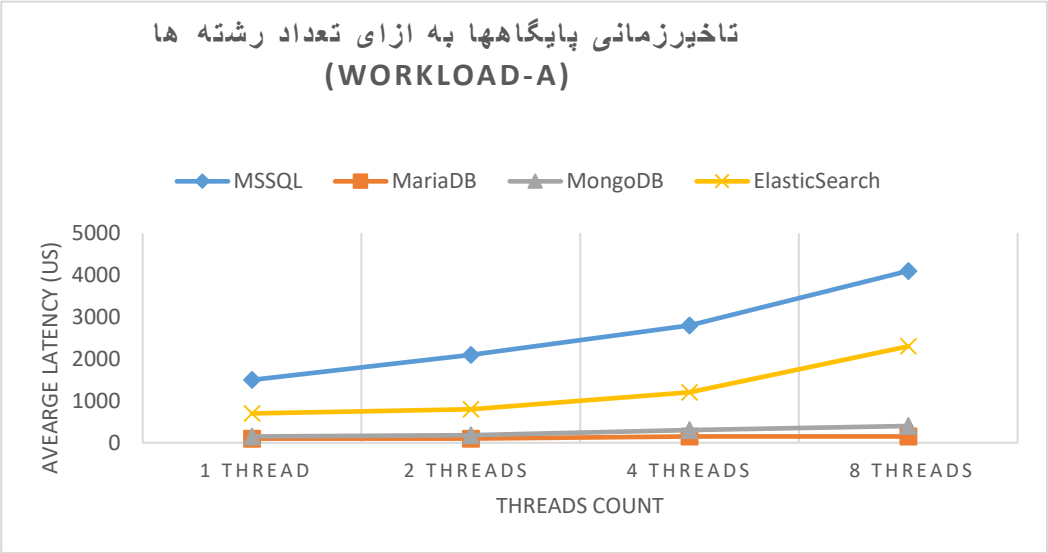
در این بخش عملکرد پایگاههای داده رابطه ای و غیر رابطه ای برای تعداد رشته های مختلف و به ازای تعداد رکورد/عملیات ده هزار بررسی شد و تعداد رشته ای که به ازای آن رفتار مناسب تری مشاهده می شود را تشخیص می دهیم.

شکلهای ۴-۱ و۴-۲ مربوط به نتایج حاصل از اجرای WorkloadA بر روی تمامی پایگاه های داده رابطه ای و غیر رابطه ای مورد بحث، برای تعداد رکورد/عملیات ده هزار و به ازای تعداد رشته های متفاوت می باشند. تعداد رشته مناسب تعدادی است که به ازای آن توانش زمانی افزایش یابد و حتی الامکان میانگین تاخیرزمانی افزایش نیابد. همانطور که در شکلهای ۴-۱و ۴-۲ قابل ملاحظه است در مجموع می توان گفت که به ازای افزایش تعداد رشته تا ۴، شاهد افزایش توانش زمانی بطور قابل ملاحظه ای بوده ایم و به نسبت تعداد رشته ۸ ، تاخیر زمانی نیز افزایش چشمگیری نداشته است و همانطور که شکلهای ۴-۳و۴-۴ نشان می دهند، این موضوع برای WorkloadB نیز صادق است. لازم به ذکر است با توجه به نتایج حاصل، در ادامه نتایج آزمایشات نیز که با بررسی جزئی تری همراه هستند به ازای تعداد ۴رشته که در کل رفتار مناسب تری نشان داده، از شکلهای ۴-۵ به بعد (تا پایان این گزارش پژوهشی) نمایش داده شده اند.

[1] Thread

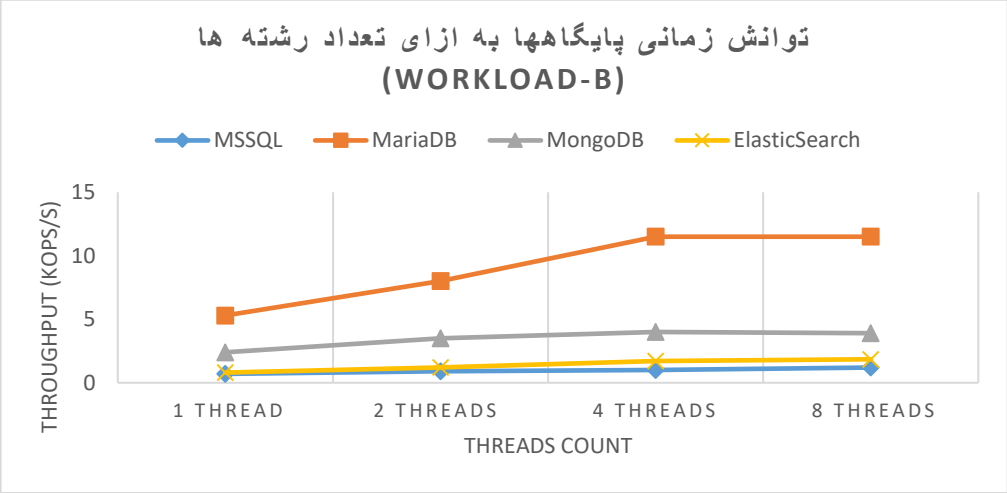


شکل۴-۱ مقایسه توانش زمانی[۱] برروی پایگاههای داده رابطه ای و غیر رابطه ای برای WorkloadA با تعداد رکورد/عملیات۱۰۰۰۰وتعداد رشته های مختلف1,2,4,8

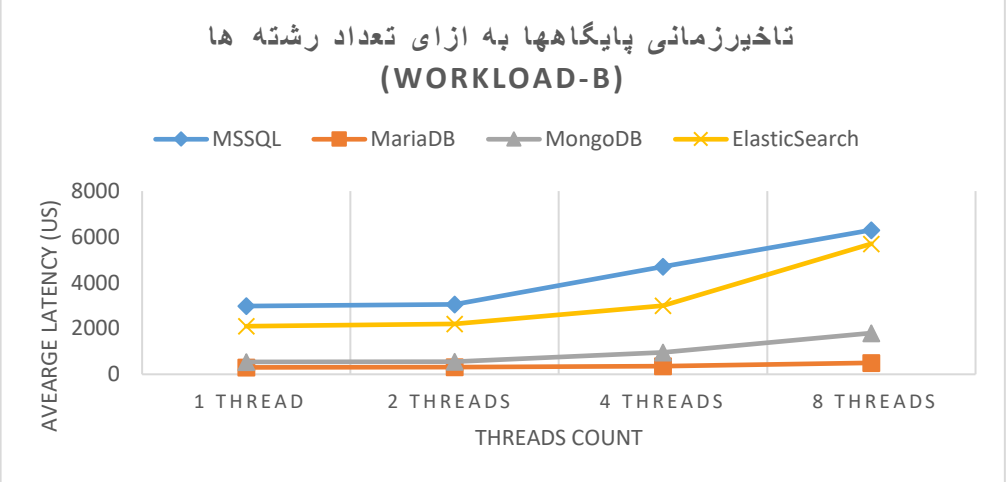


شکل ۴-۲ مقایسه تاخیر زمانی[۲] بر روی پایگاههای داده رابطه ای و غیر رابطه ای برای WorkloadA با تعداد رکورد/عملیات ۱۰۰۰۰ و تعداد رشته های مختلف1,2,4,8

[۱] Throughput: توانش زمانی یا تعداد عملیات انجام شده در واحد زمان (KOps/Sec: Kiloo-Operations/ Second)



شکل۴-۳ مقایسه توانش زمانی بر روی پایگاههای داده رابطه ای و غیر رابطه ای برای WorkloadB با تعداد رکورد/عملیات ۱۰۰۰۰و تعداد رشته های مختلف1,2,4,8



شکل ۴-۴ مقایسه تاخیر زمانی بر روی پایگاههای داده رابطه ای و غیر رابطه ای برای WorkloadB با تعداد رکورد/عملیات ۱۰۰۰۰ و تعداد رشته های مختلف1,2,4,8

از دیدگاه دیگر با بررسی عملکرد کلی سیستمها مشخص است که توانش زمانی بر روی پایگاههای داده رابطه ای و غیر رابطه ای برای هر دو WorkloadA و WorkloadB با تعداد رکورد/عملیات ده هزار و تعداد رشته های مختلف 1,2,4,8 به اختصار به شرح زیر است.

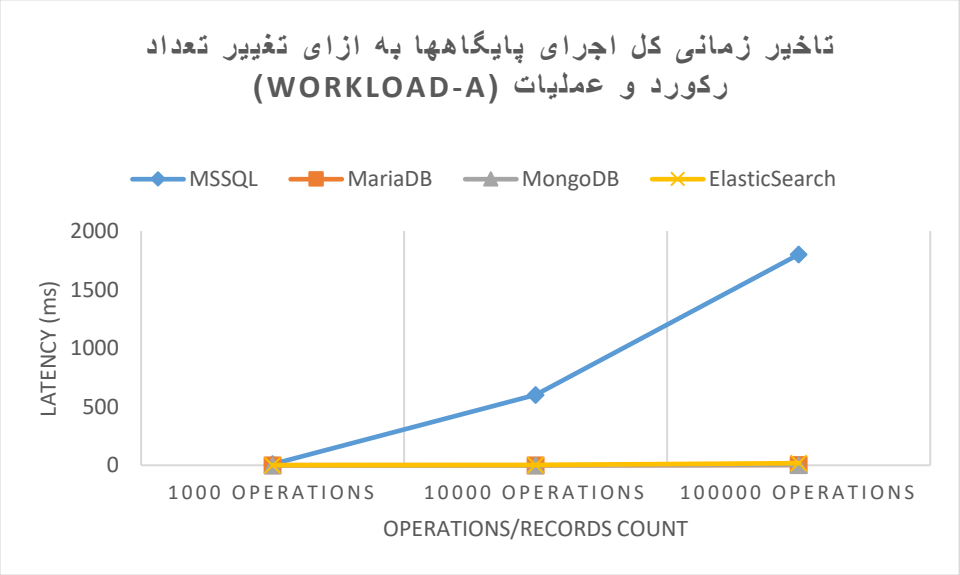
Throughput: MariaDB > MongoDB> ElasticSearch> MSSQL

Average Latency: MariaDB < MongoDB< ElasticSearch< MSSQL

به بیان بهتر توانش زمانی برای MariaDB در کلیه رشته های مختلف در بهترین وضعیت، MongoDB در درجه دوم، ElaticSearch در درجه سوم وMSSQL در درجه آخر قرار دارد. و مسلما میزان تاخیر زمان بازیابی در MariaDB در بهترین وضعیت با کمترین میزان تاخیر در بازیابی است، MongoDB در رده دوم و ElasticSearch در درجه بعدی و MSSQL در بدترین حالت و با بیشترین میزان تاخیر در بازیابی در آخرین رده قرار می گیرد.

۴-۲-۲ بررسی عملکرد پایگاههای داده رابطه ای و غیر رابطه ای با افزایش تعداد عملیات/رکورد-نرخ افزایش زمان کل اجرای بارکاری

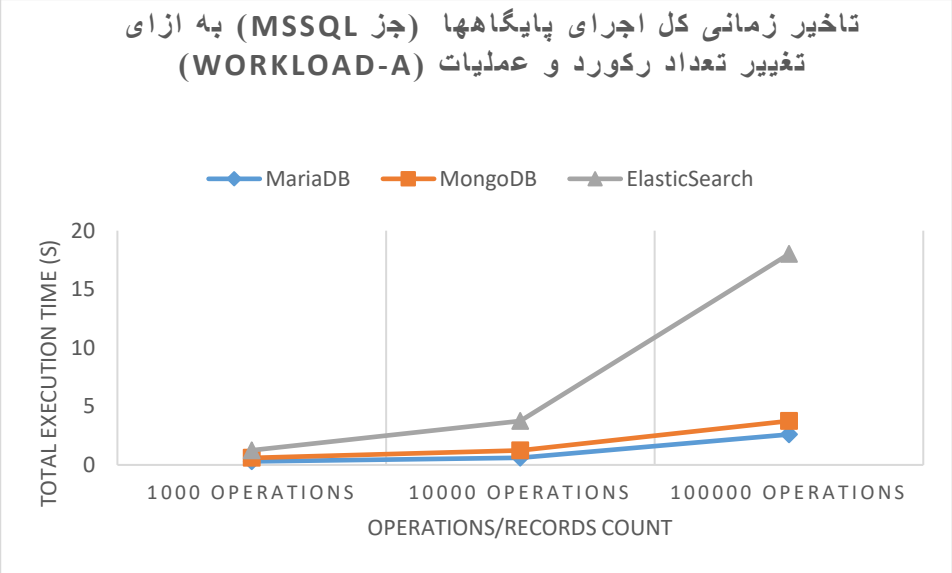
در این بخش عملکرد پایگاههای داده رابطه ای و غیر رابطه ای به ازای تعداد رکورد/عملیات ۱۰۰۰، ۱۰۰۰۰ و ۱۰۰۰۰۰ با در نظر گرفتن نرخ افزایش زمان کل اجرای بارکاری مورد بررسی قرار می گیرد. با توجه به رابطه معکوس بین توانش زمانی و تاخیر زمانی از هم اکنون به بعد در نتایج آزمایشات گزارش شده صرفا به نمایش نمودار تاخیر زمانی که در واقع یک پارامتر بسیار مهم در بررسی عملکرد پایگاههای داده (با توجه به سرعت بازیابی اطلاعات[1] در حین جستجوست) خواهیم پرداخت.



شکل ۴-۵ زمان کل اجرای[2] WorkloadA برروی پایگاههای داده رابطه ای و غیر رابطه ای با تعداد رکورد/عملیات ۱۰۰۰، ۱۰۰۰۰ و ۱۰۰۰۰۰ (الف)

[1] Information Retrival
[2] Total Excution Time

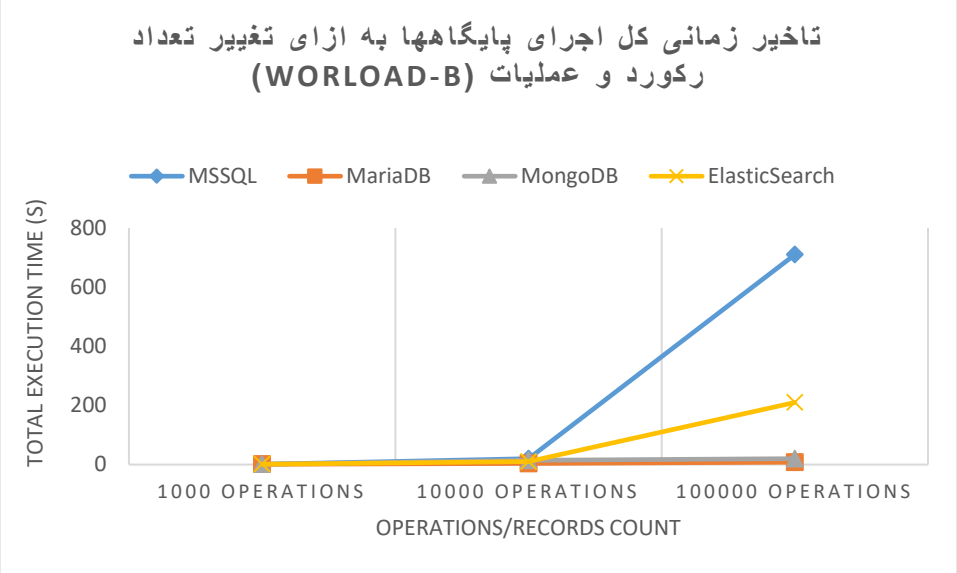
شکل۴-۵ رفتار پایگاه های داده مورد بحث را در برابر افزایش تعداد عملیات رکورد برای WorkloadA(-خوانش ۱۰۰٪) نشان می دهد. نکته ی قابل توجه در این نمودار نرخ افزایش شدید زمان کل اجرای عملیات با افزایش ده هزار به صد هزار عملیات/رکورد برای SQL Server را نشان می دهد، تا حدی که بررسی رفتار سایر پایگاه های داده و مقایسه آن ها با یکدیگر را دشوار ساخته است. به همین علت در شکل۴-۶ برای پرداختن به این مقایسه SQL Server را حذف کرده است.



شکل۴-۶ تاخیر زمانی کل اجرای WorkloadA بر روی پایگاههای داده رابطه ای و غیر رابطه ای با تعداد رکورد/عملیات ۱۰۰۰، ۱۰۰۰۰ و ۱۰۰۰۰۰(ب)

همانگونه که در شکل۴-۶ قابل مشاهده است، نرخ افزایش زمان اجرای بارکاری برای پایگاه های داده MongoDB و MariaDB تقریبا مشابه است اما برای Elasticsearch این نرخ به طرز قابل توجهی بیشتر است. در مجموع می توان گفت در برابر افزایش تعداد رکورد/عملیات برای WorkloadA (۱۰۰٪ خوانش)، بدترین رفتار مربوط به پایگاه داده SQL Server و در درجه بعدی

مربوط به Elasticsearch می‌باشد و پایگاه های داده MongoDB و MariaDB به نسبت این دو رفتار با ثبات تری از خود نشان می دهند. همین موضوع همچنان که در شکل ۴-۷ قابل مشاهده است برای WorkloadB (۱۰۰٪ اسکن) نیز صدق می کند.



شکل۴-۷ زمان کل اجرای WorkloadB بر روی پایگاههای داده رابطه ای و غیر رابطه ای با تعداد رکورد/عملیات ۱۰۰۰، ۱۰۰۰۰ و ۱۰۰۰۰۰

۴-۲-۳ بررسی عملکرد پایگاههای داده رابطه ای و غیر رابطه ای با افزایش تعداد عملیات/رکورد –توانش زمانی و تاخیر زمانی

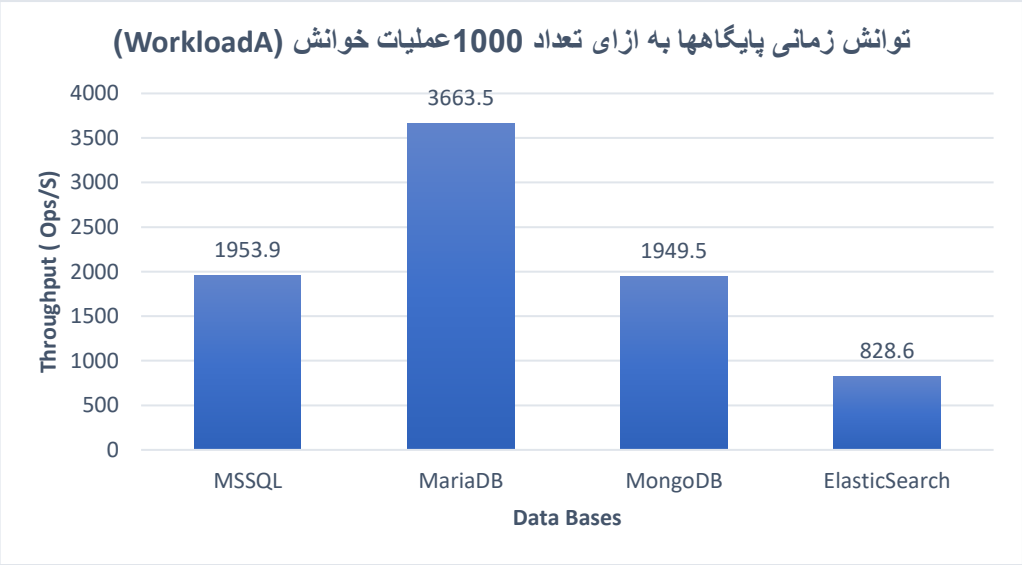
در ادامه به صورت جزئی تر رفتار پایگاه های داده مورد بحث را به ازای تعداد رکورد/عملیات ۱۰۰۰، ۱۰۰۰۰ و ۱۰۰۰۰۰ در قیاس با یکدیگر با در نظر گرفتن توانش زمانی و تاخیر زمانی می سنجیم:

همانطور که دو شکل ۴-۸ و ۴-۹ نتایج حاصل از WorkloadA را نشان می دهند بهترین رفتار مربوط به پایگاه داده MariaDB است که بالاترین توانش زمانی و کمترین تاخیر زمانی را ثبت کرده است و در مقابل آن پایگاه داده Elasticsearch قرار می گیرد که کمترین توانش زمانی و بالاترین تاخیر زمانی را ثبت کرده است. آمار ثبت شده توسط پایگاه های داده MongoDB و SQL Server نزدیک به هم می‌باشد با این تفاوت که MongoDB آمار بهتری را برای Latency ثبت کرده است. (خصوصا برای ۹۹ امین صدک تاخیر[1] که مربوط به ۱ درصد درخواست هایی است که بدترین تاخیر زمانی را تجربه کردند و این نشان از ثبات بیشتر MongoDB دارد).

دو شکل ۴-۱۰ و ۴-۱۱ که نتایج حاصل از WorkloadB را نشان می دهند نیز کلیات تحلیل مربوط به WorkloadA صادق است با این تفاوت که با مقایسه دقیق تر عملکرد پایگاه های داده MongoDB و SQL Server می توان دریافت که برتری MongoDB نسبت به SQL Server در WorkloadB (۱۰۰ ٪ اسکن) مشهود تر است.

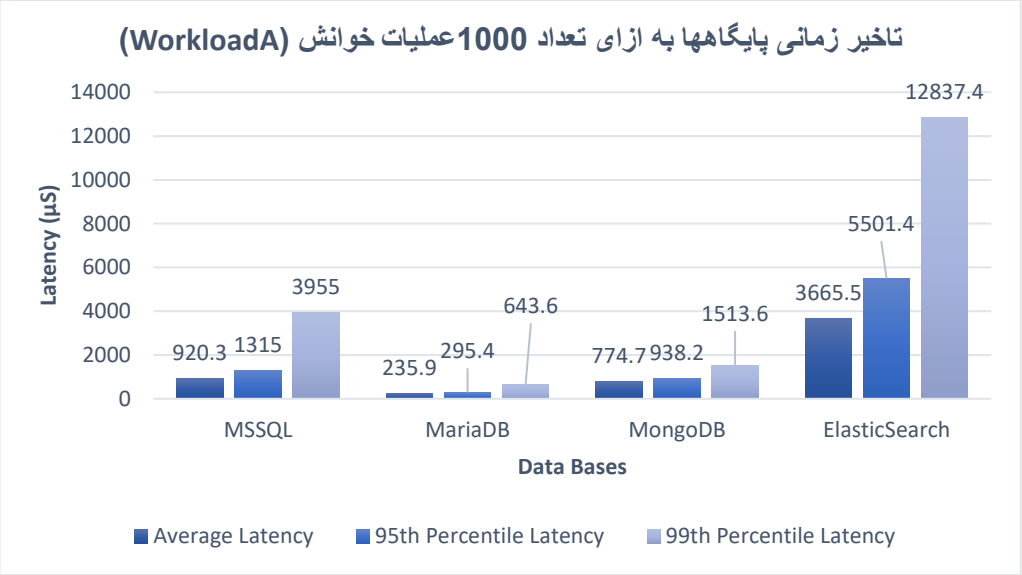
[1] 99th Percentile Latency

تعداد عملیات/رکورد۱۰۰۰



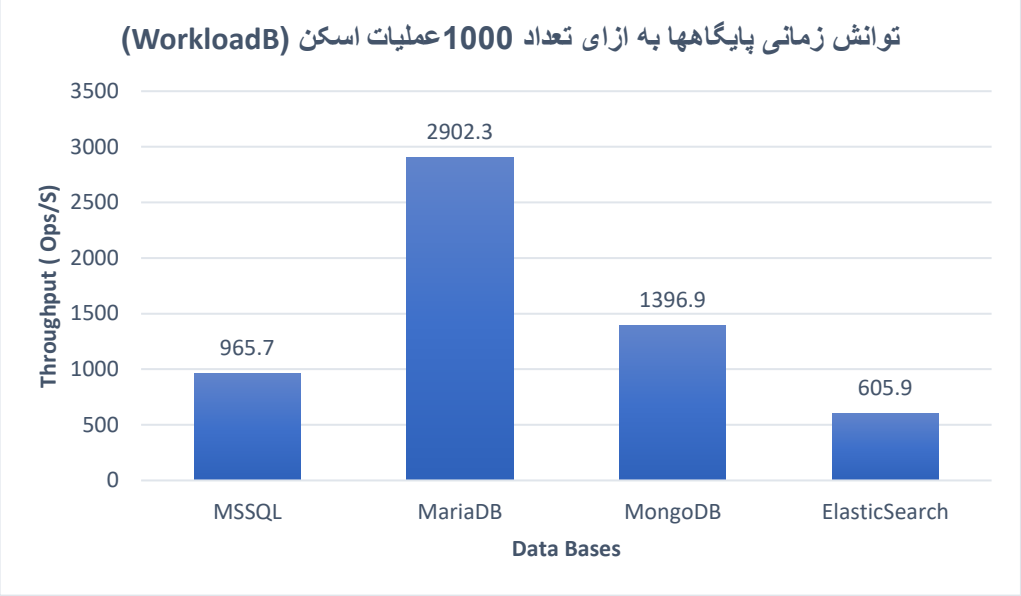
شکل۴-۸ مقایسه توانش زمانی بر روی پایگاههای داده رابطه ای و غیر رابطه ای برای WorkloadA

با تعداد رکورد/عملیات ۱۰۰۰

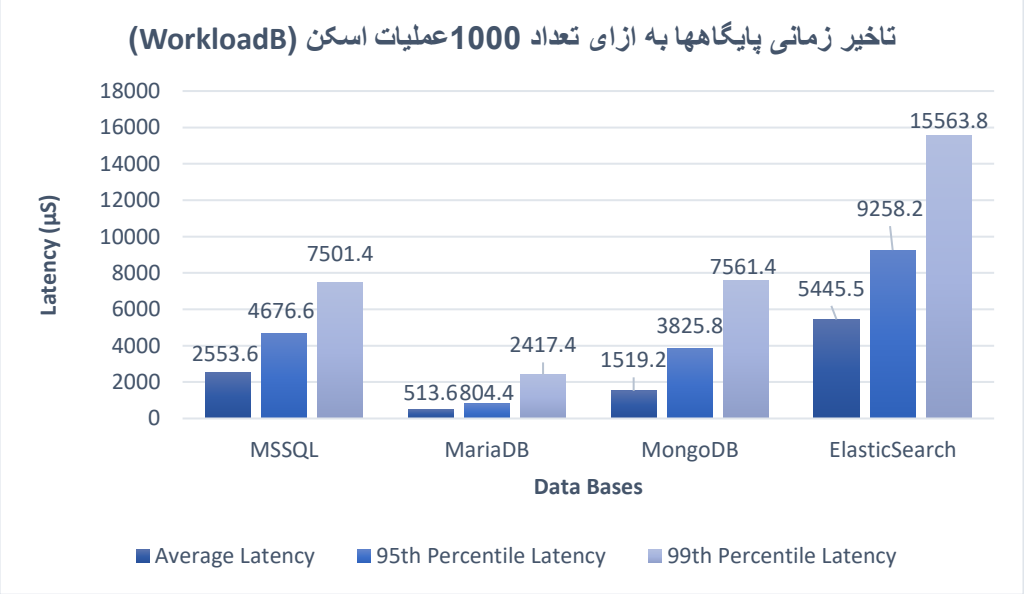


شکل۴-۹ مقایسه تاخیر زمانی بر روی پایگاههای داده رابطه ای و غیر رابطه ای برای WorkloadA

با تعداد رکورد/عملیات۱۰۰۰

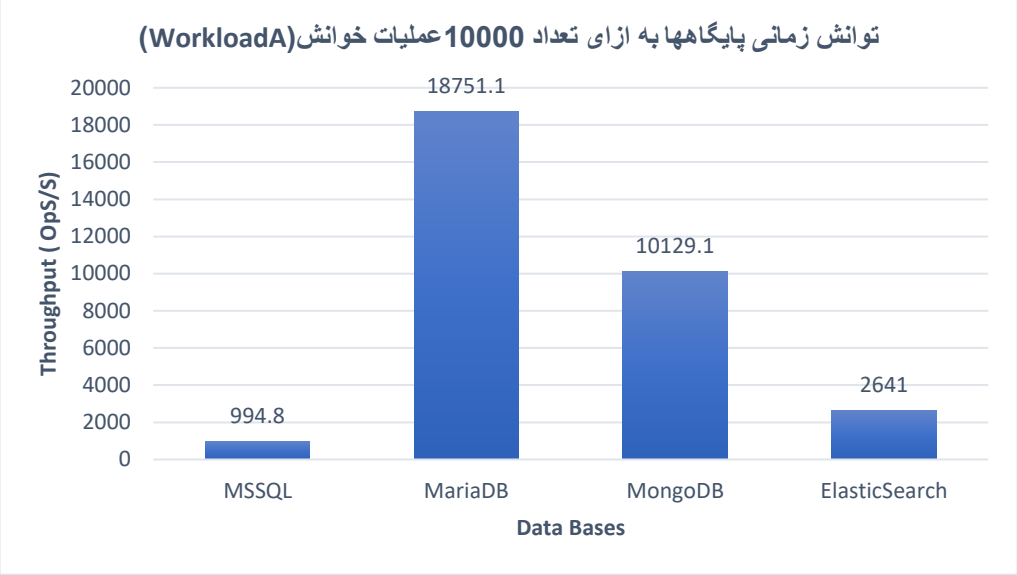


شکل۴-۱۰ مقایسه توانش زمانی بر روی پایگاههای داده رابطه ای و غیر رابطه ای برای WorkloadB با تعداد رکورد/عملیات ۱۰۰۰

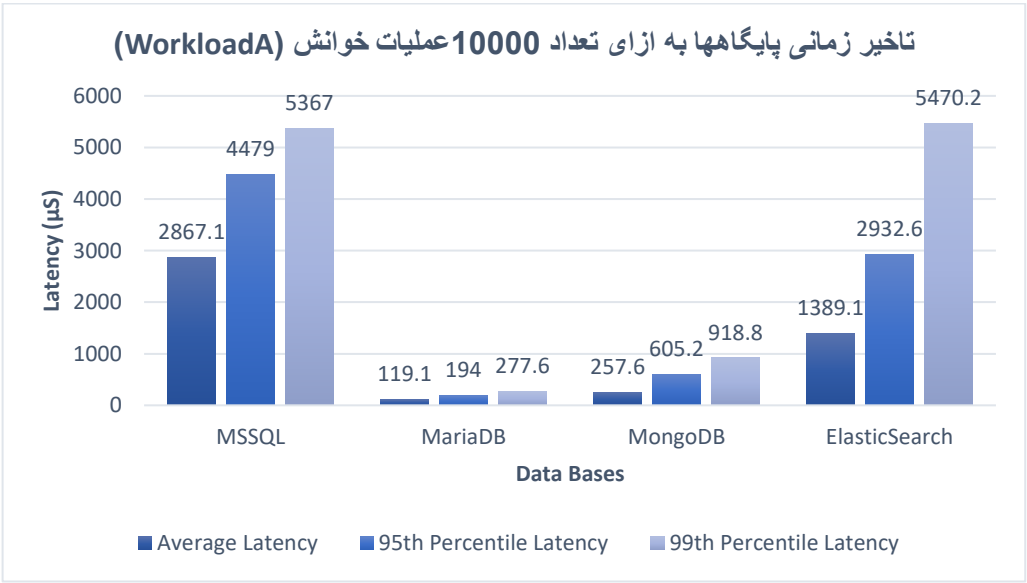


شکل۴-۱۱مقایسه تاخیر زمانی بر روی پایگاههای داده رابطه ای و غیر رابطه ای برای WorkloadB با تعداد رکورد/عملیات ۱۰۰۰

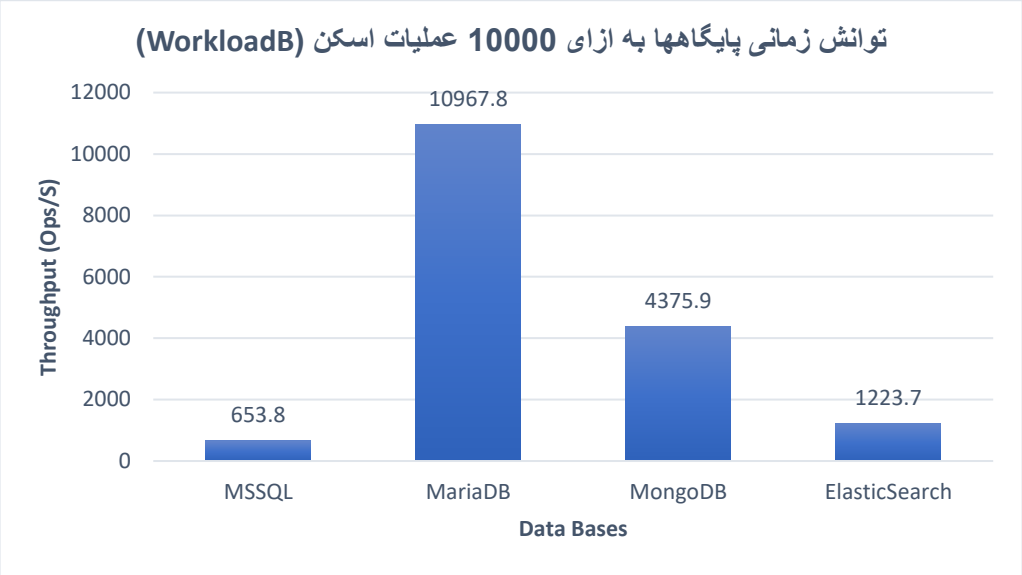
تعداد عملیات/رکورد۱۰۰۰۰



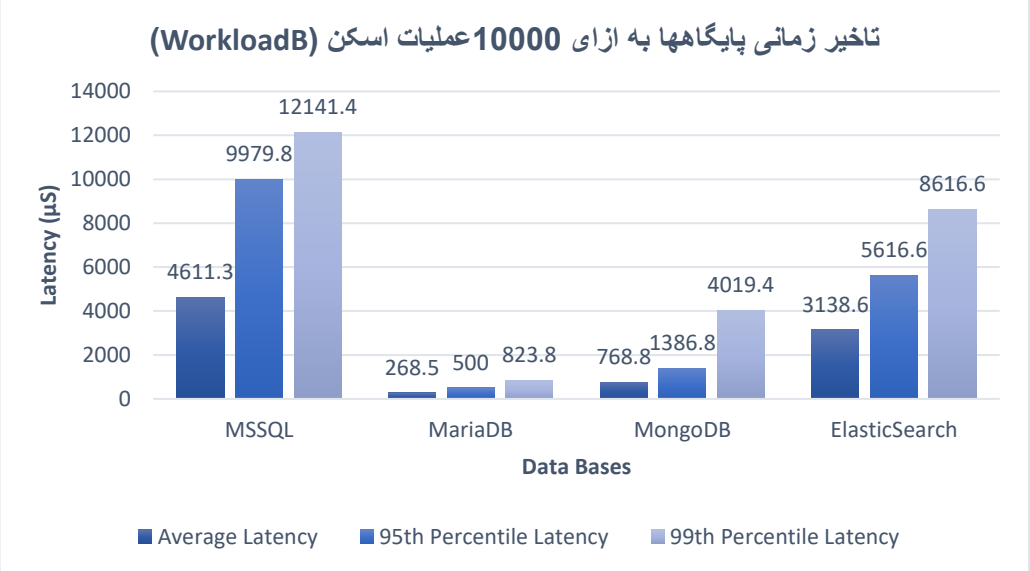
شکل۴-۱۲ مقایسه توانش زمانی بر روی پایگاههای داده رابطه ای و غیر رابطه ای برای WorkloadA با تعداد رکورد/عملیات ۱۰۰۰۰



شکل ۴-۱۳مقایسه تاخیر زمانی بر روی پایگاههای داده رابطه ای و غیر رابطه ای برای WorkloadA با تعداد رکورد/عملیات ۱۰۰۰۰



شکل۴-۱۴ مقایسه توانش زمانی بر روی پایگاههای داده رابطه ای و غیر رابطه ای برای WorkloadB با تعداد رکورد/عملیات ۱۰۰۰۰

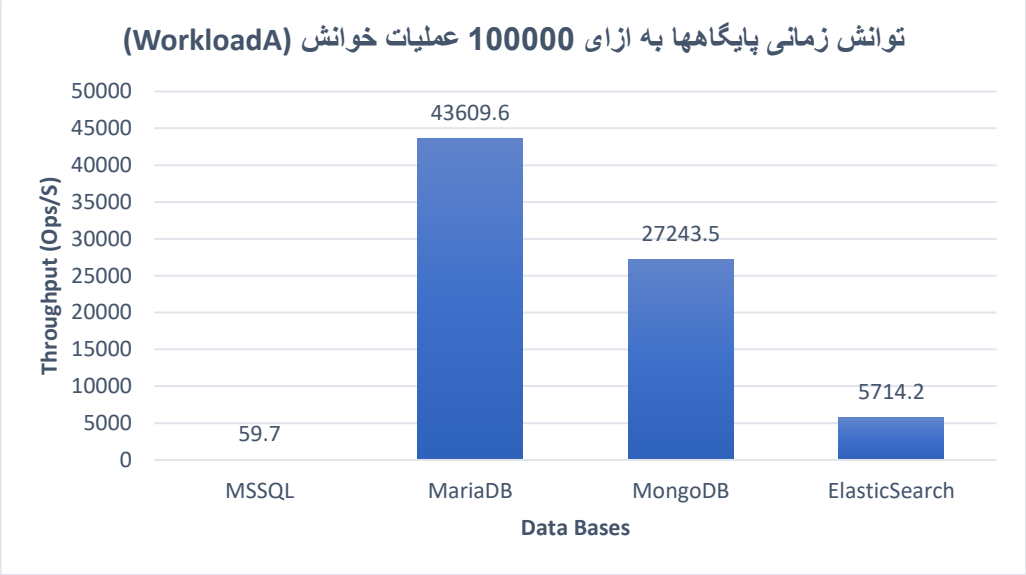


شکل ۴-۱۵ مقایسه تاخیر زمانی بر روی پایگاههای داده رابطه ای و غیر رابطه ای برای WorkloadB با تعداد رکورد/عملیات ۱۰۰۰۰

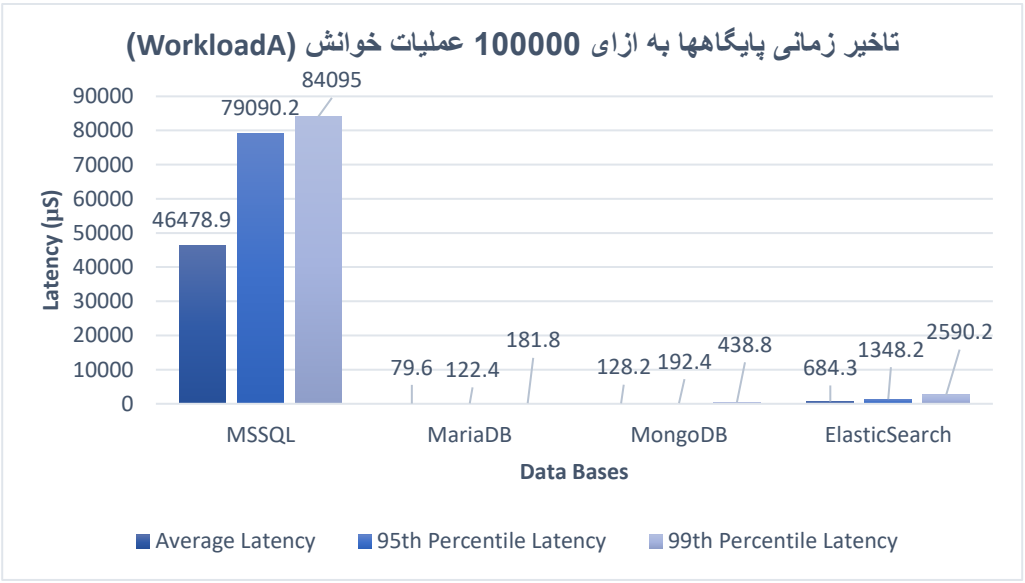
با مقایسه شکل های ۴-۱۲ و ۴-۱۳ با شکل های ۴-۸ و ۴-۹ می توان دریافت که با افزایش تعداد عملیات/رکورد ها از ۱۰۰۰ به ۱۰۰۰۰، SQL Server افت شدیدی داشته و به نسبت آن Elasticsearch عملکرد بهتری از خود نشان می دهد و برخلاف SQL Server، توانش زمانی آن افزایش و تاخیر زمانی آن کاهش یافته است. بنابراین با افزایش تعداد درخواست ها به ده هزار در این قسمت، بدترین رفتار مربوط به SQL Server است. بهترین عملکرد کماکان مربوط به پایگاه داده MariaDB است و پس از آن پایگاه داده MongoDB قرار می گیرد. با مقایسه شکل های ۴-۱۴ و ۴-۱۵ که (مربوط به WorkloadB) با شکل های ۴-۱۰ و ۴-۱۱ نیز همین نتایج حاصل می شود.

شکل های ۴-۱۶ تا ۴-۱۹ نشان می دهند که با افزایش تعداد عملیات/رکورد ها به ۱۰۰۰۰۰ روند تغییراتی که در قسمت قبلی (مربوط به تعداد عملیات/رکورد ها به ۱۰۰۰۰) توضیح داده شد در این قسمت نیز تکرار می شود و SQL Server به روند نزولی عملکرد خود را با شدت بیشتری ادامه می دهد.

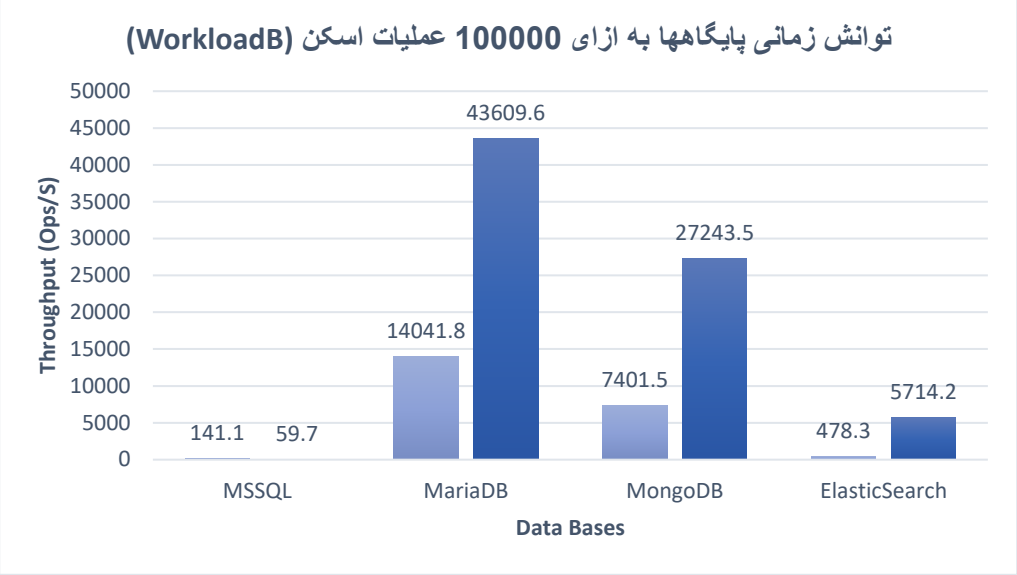
تعداد عملیات/رکورد۱۰۰۰۰۰



شکل ۴-۱۶ مقایسه توانش زمانی بر روی پایگاههای داده رابطه ای و غیر رابطه ای برای WorkloadA با تعداد رکورد/عملیات ۱۰۰۰۰۰

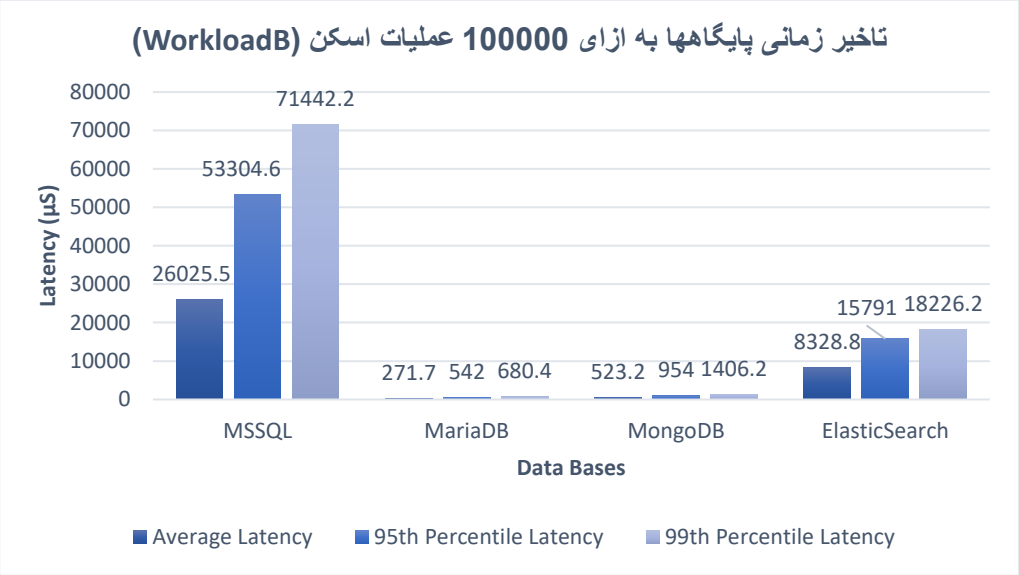


شکل ۴-۱۷مقایسه تاخیر زمانی بر روی پایگاههای داده رابطه ای و غیر رابطه ای برای WorkloadA با تعداد رکورد/عملیات ۱۰۰۰۰۰



شکل۴-۱۸مقایسه توانش زمانی بر روی پایگاههای داده رابطه ای و غیر رابطه ای برای WorkloadB

با تعداد رکورد/عملیات ۱۰۰۰۰۰



شکل۴-۱۹ مقایسه تاخیر زمانی بر روی پایگاههای داده رابطه ای و غیر رابطه ای برای WorkloadB

با تعداد رکورد/عملیات ۱۰۰۰۰۰

۴-۲-۴ جمع بندی نتایج مربوط به معیار ارزیابی سرویس ابری یاهو!(YCSB)

در مقیاسی که پایگاه های داده SQL Server، MariaDB، MongoDB و Elasticsearch را بوسیله YCSB و برای بارهای کاری ۱۰۰٪ خوانش و ۱۰۰٪ اسکن مورد ارزیابی قرار دادیم، حکایت از برتری چشمگیر MongoDB و MariaDB نسبت به دو پایگاه داده دیگر داشت. همچنین مشاهده شد که با افزایش تعداد عملیات/رکورد ها SQL Server برتری خود را نسبت به Elasticsearch به طور چشمگیری از دست می دهد و در بین ۴ پایگاه داده مورد بحث ضعیف ترین عملکرد را به خود اختصاص می دهد.

۴-۳ بررسی پایگاههای داده رابطه ای و غیر رابطه ای در جستجوی تمام متن

در این قسمت به بررسی عملکرد جستجوی تمام متن برای پایگاه‌های داده SQL Server ، MariaDB، MongoDB و Elasticsearch می پردازیم. برای ارزیابی این عملکرد چهار نوع پرس و جوهای آزمایشی در نظر گرفته شده است که به شرح زیر می باشند:

- Single: به جستجوی یک کلمه مشخص در پایگاه داده می پردازد.
- OR: به جستجوی سه کلمه در پایگاه داده می پردازد بدین صورت که هر رکورد که حداقل یک کلمه از سه کلمه را در متن خود داشته باشد جزء نتایج پذیرفته شود.
- AND: به جستجوی سه کلمه در پایگاه داده می پردازد بدین صورت که هر رکورد پذیرفته شده بایستی هر سه کلمه را در متن خود داشته باشد.
- Exact Phrase: به جستجوی یک عبارت مشخص (که متشکل از سه کلمه می‌باشد) در پایگاه داده می پردازد.

برای ساخت یک دیتاست با عنوان NoSQL-Dataset، از روش نمونه گیری تصادفی از ۶ میلیون چکیده لاتین موجود درمخزن داده های مرکز منطقه ای استفاده شده است. شایان ذکر است که در سال اخیر رشد چشمگیر بالغ بر ۵۰٪ی در تعداد رکوردهای لاتین در دو پایگاه مقالات انگلیسی[1] و مقالات انگلیسی مهندسی[2] شاهد بودیم. با توجه به خصوصیت جستجوی تمام متن، هر رکورد به صورت تقریبی حدود ۳۰۰ کلمه است و داده های مورد بررسی بالغ بر یک میلیارد و هشتصد میلیون می باشد که داده حجیم محسوب می شود.

[1] English Articles
[2] Engineering English Articles

به منظور بررسی عملکرد جستجوی تمام متن در انواع پرس و جوهای آزمایشی[1] فوق؛ برای پرس و جوی آزمایشی single هزار کلمه، برای کوئیری OR هزار گروه سه کلمه ای، برای پرس و جوهای آزمایشی AND هزار گروه سه کلمه ای دیگر و برای پرس و جوهای آزمایشی Exact Phrase هزارعبارت معنادار سه کلمه ای به صورت تصادفی از دیتاست NoSQL-Dataset استخراج شده و بر روی پایگاههای داده SQL Server، MariaDB، MongoDB وElasticsearch اجرا شده است.

دو معیار برای مقایسه پایگاه های داده انتخاب شده اند معیار اول زمان بازیابی است. این زمان عبارت است از مجموع مدت زمانی که صرف یافتن تمامی رکوردهایی که دارای ویژگی مورد نظر باشند، مرتب کردن آن‌ها بر اساس امتیاز[2] (relevance score)، ذخیره بیست رکورد اول (بالاترین امتیاز) در یک فایل و محاسبه تعداد کل این رکوردهاست. متوسط زمان بازیابی سه بار اجرای هر یک از پرس و جوهای آزمایشی (۳۰۰۰ پرس و جوهای آزمایشی) به عنوان نتیجه نهایی زمان بازیابی در گزارشات آمده است.

معیار دوم کیفیت بازیابی نتایج است که به مقایسه بهترین نتیجه بازیابی شده در ۴ نوع پرس و جوهای آزمایشی می پردازد. به این منظور روابط پیشنهادی QRRT در این پژوهش در فرمولهای ۴-۱ الی ۴-۴ ارائه شده است که قابلیت مقایسه کیفیت نتایج بازیابی شده بین پایگاه های مختلف با توجه به نوع پرس و جوهای آزمایشی را می دهد که در قسمت ۴,۳,۲,۳ بتفصیل به آن پرداخته شده است.

[1] Query
[2] Relevance Score

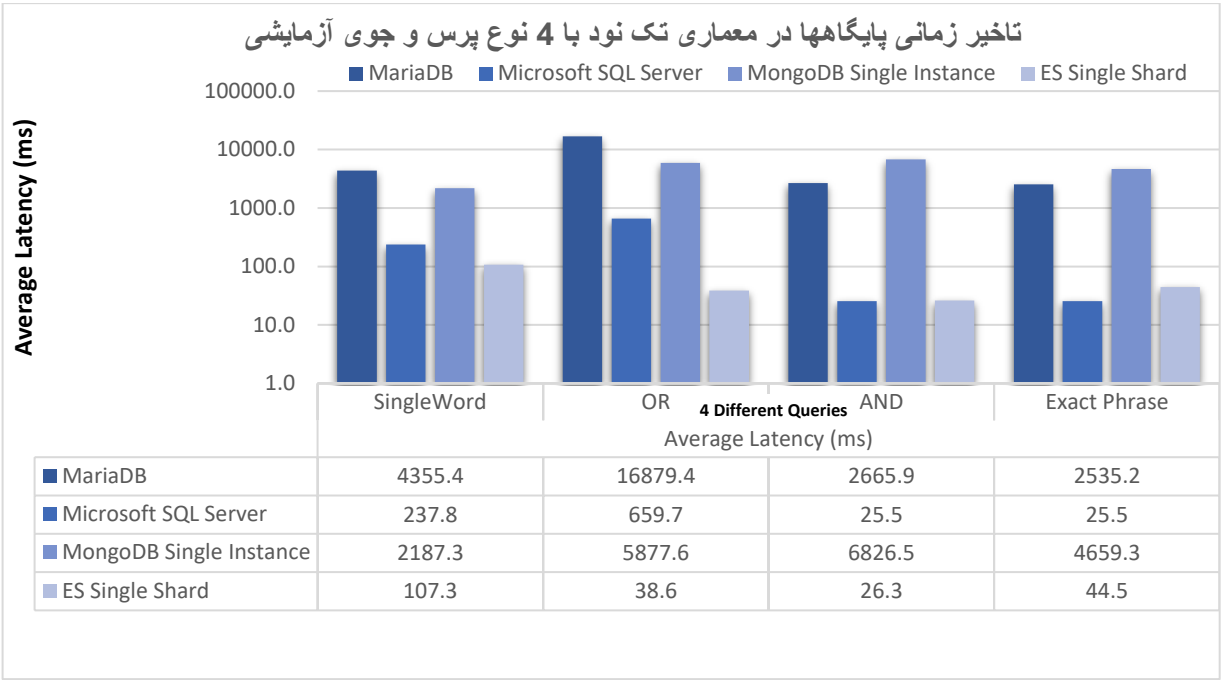
تمام آزمایشات برای پایگاههای داده غیر رابطه ای که قابلیت کلاستر /خوشه بندی[1] را دارند در سه حالت تک نود[2]،کلاستر سه نود شارد خرد شده[3] و کلاستر شش نود شارد خرد شده[4] اجرا و تاثیر قابلیت خوشه بندی که از مزایای پایگاههای داده غیر رابطه ای محسوب می شود گزارش شده است.

گفتنی است پیش از اجرای پرس و جوهای آزمایشی لازم است تا فرآیند ایندکسینگ تمام متن بر روی مجموعه رکوردها در پایگاهها انجام شده باشد. برای اینکه ویژگی‌های این فرآیند حتی الامکان در چهار پایگاه داده مورد بحث یکسان باشد، ویژگی‌های افزوده‌ای همچون ریشه یابی[5] و حذف کلمه های ایستا[6] از فرآیند نمایه سازی ایندکسینگ، که برای پایگاه داده‌های مختلف به صورت‌های مختلفی اجرا می شود، از تنظیمات پیش فرض پایگاه‌های داده مورد بحث، حذف شده است.

[1] Sharded Cluster
[2] Signle Node
[3] 3-Sharded Cluster
[4] 6-Sharded Cluste
[5] Stemming
[6] Stopword

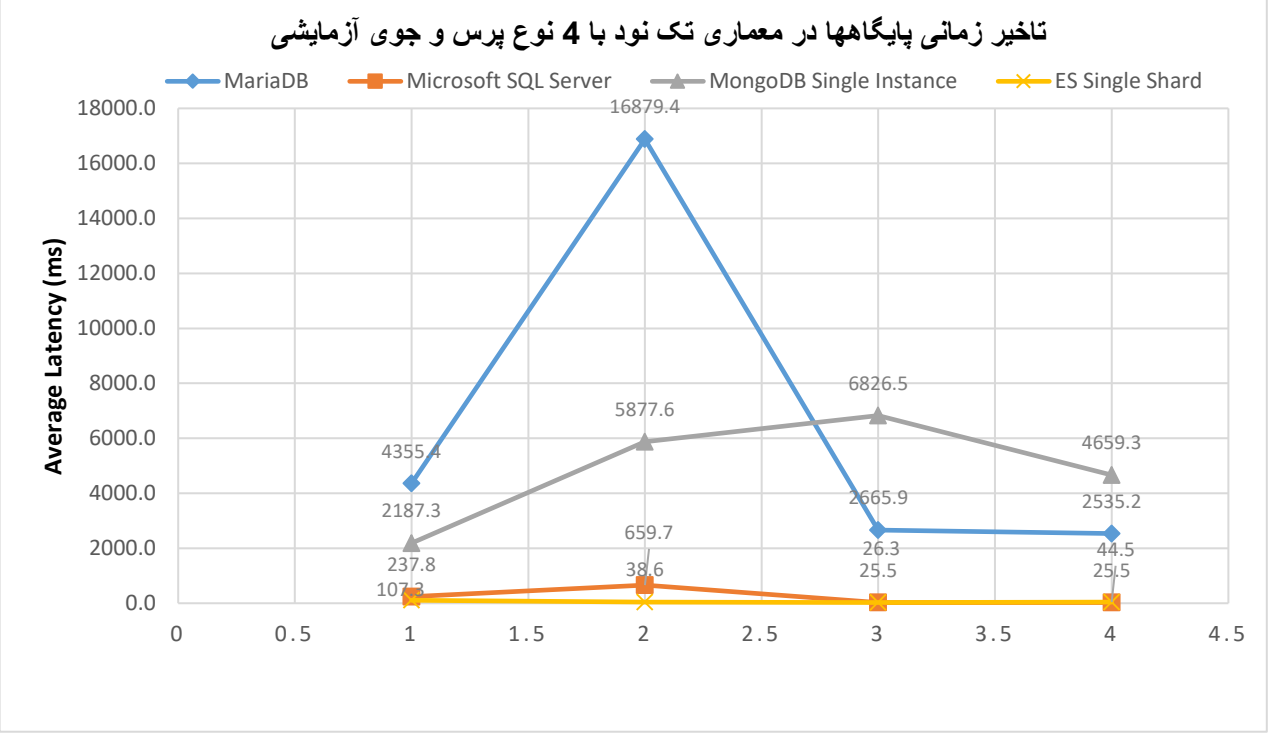
۴–۳–۱ بررسی عملکرد پایگاههای داده رابطه ای وغیر رابطه ای در جستجوی تمام متن بدون خوشه بندی (Single Node) با معیار زمان بازیابی

شکل ۴-۲۰ و ۴-۲۱ به مقایسه عملکرد جستجو و بازیابی تمام متن در پایگاه‌های داده رابطه ای و غیر رابطه ای در حالت ساده یک تک نود (single node) یا به اصطلاح بدون خوشه بندی بر روی چهار نوع پرس و جوهای آزمایشی می پردازد.



Average Latency (ms)	SingleWord	OR	AND	Exact Phrase
MariaDB	4355.4	16879.4	2665.9	2535.2
Microsoft SQL Server	237.8	659.7	25.5	25.5
MongoDB Single Instance	2187.3	5877.6	6826.5	4659.3
ES Single Shard	107.3	38.6	26.3	44.5

شکل ۴-۲۰ مقایسه تاخیر زمانی بر روی پایگاههای داده رابطه ای و غیر رابطه ای بدون خوشه بندی با چهار نوع پرس و جوهای آزمایشی - در مقیاس لگاریتمی



شکل ۴-۲۱ مقایسه تاخیر زمانی بر روی پایگاههای داده رابطه ای و غیر رابطه ای بدون خوشه بندی با چهار نوع پرس و جوهای آزمایشی (نمایش مقایسه ای دیگری از نوسانات داده های ارائه شده در شکل ۴- ۲۰ در حالت غیر لگاریتمی)

شکل ۴-۲۰ و ۴-۲۱ به مقایسه عملکرد جستجوو بازیابی تمام متن در پایگاه‌های داده رابطه ای و غیر رابطه ای در حالت ساده یک تک نود (single node) یا به اصطلاح بدون خوشه بندی بر روی چهار نوع پرس و جوهای آزمایشی می پردازد.

در ابتدا نکته قابل توجه این شکل، طولانی تر بودن اجرای پرس و جوهای آزمایشی نوع OR است. این موضوع نشان می دهد که هر چه تعداد رکورد های نتیجه پرس و جوهای آزمایشی بیشتر باشد، سربار محاسبه امتیاز آن ها و مرتب کردن آن ها بر اساس این امتیاز بیشتر خواهد بود و همین امر باعث می شود زمان اجرای پرس و جوهای آزمایشی نوع OR در این پایگاه‌های داده به مراتب بیشتر از سایر پرس و جوهای آزمایشی باشد.

مقایسه عملکرد جستجو و بازیابی تمام متن در پایگاه‌های داده رابطه‌ای Microsoft SQL Serverو MariaDB بدون خوشه بندی حاکی از آن است که SQL Server در هر چهار نوع پرس و جوهای آزمایشی با برتری بسیار قابل چشمگیری قادر به ثبت میانگین تاخیر کمتری نسبت به MariaDB بوده است و بعبارت دیگر از نظر سرعت عملکردی کارایی بهتری را از خود نشان داده است.

به همین نحو مقایسه عملکرد جستجو و بازیابی تمام متن در پایگاه‌های داده غیررابطه‌ای ElasticSearchو MongoDB بدون خوشه بندی حاکی از آن است که ElasticSearch در هر چهار نوع پرس و جوهای آزمایشی با برتری بسیار قابل چشمگیری قادر به ثبت میانگین تاخیر کمتری نسبت به MongoDB بوده است و بعبارت دیگر از نظر سرعت عملکردی کارایی بهتری را از خود نشان داده است.

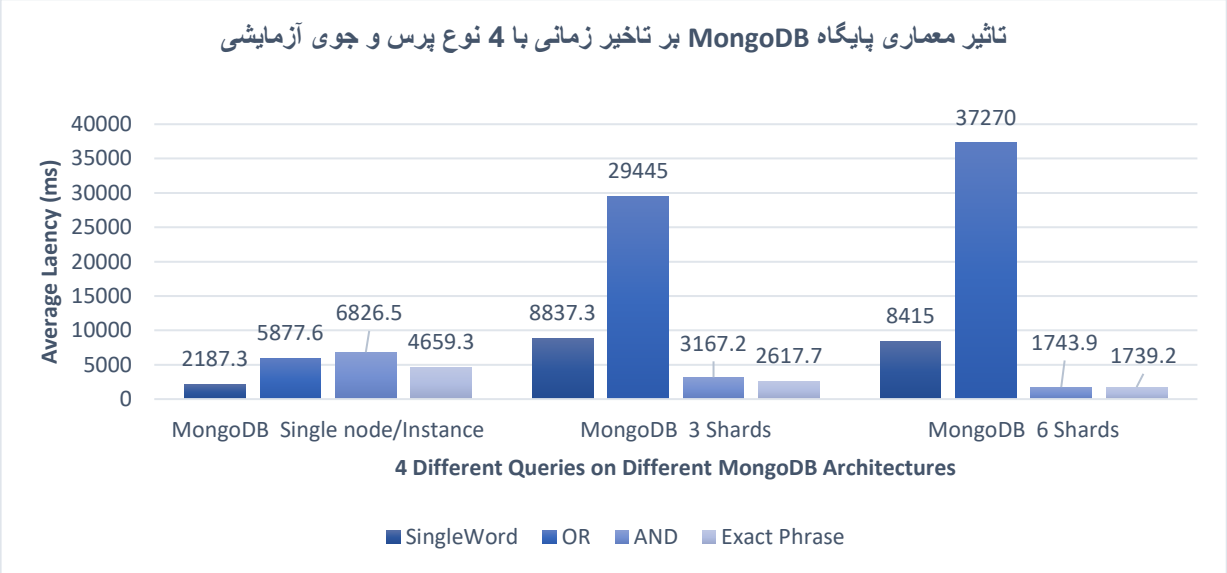
همانطورکه درشکلهای ۴-۲۰ و ۴-۲۱ نمایش داده شده است، بطور کلی در حالتی که معماری بدون خوشه بندی مطرح است، در اکثر مواقع در هر چهار نوع پرس و جوهای آزمایشی برتری عملکرد بسیار قابل توجه ElasticSearchو Microsoft SQL Server نسبت به دو نمونه دیگر پایگاه داده MariaDBو MongoDB مشهود است.

۴-۳-۲ بررسی عملکرد پایگاههای داده رابطه ای وغیر رابطه ای در جستجوی تمام متن با قابلیت خوشه بندی (Sharded Cluster)

در این قسمت به بررسی دقیقتر تاثیر اضافه نمودن شاردینگ بر دو پایگاه داده غیررابطه ای MongoDB و ElasticSearch که از قابلیتهای مهم آنهاست می پردازیم. بدین منظور دو حالت 3 Shard و 6 Shard در نظر گرفته می شود و در قیاس با حالت تک نود [1] نتایج بررسی ها برای هر ۴ نوع پرس و جوهای آزمایشی اعلام میشود.

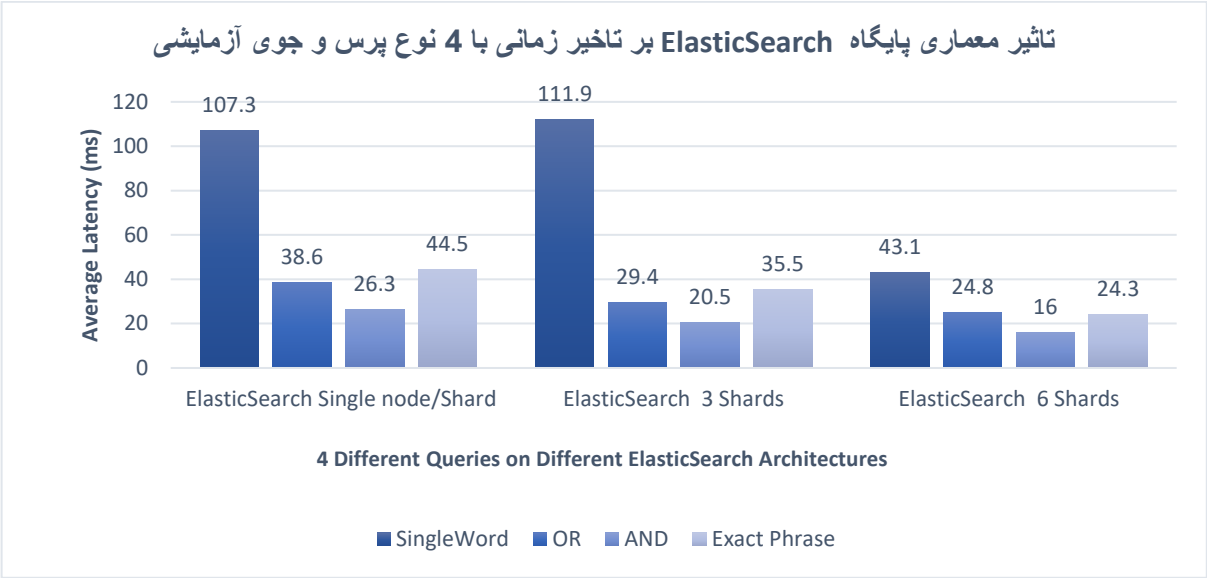
۴-۳-۲-۱ بررسی عملکرد پایگاههای داده غیر رابطه ای با در نظر گرفتن قابلیت خوشه بندی (Sharded Cluster) به تفکیک معماری با معیار زمان بازیابی

در این قسمت به بررسی تاثیر شاردینگ در عملکرد جستجوی تمام متن Full-Text Search برای پایگاه داده غیررابطه ای MongoDB و ElasticSearch می پردازیم.



شکل ۴-۲۲ مقایسه تاخیر زمانی پایگاه داده غیر رابطه ای MongoDB با در نظر گرفتن شاردینگ با چهار نوع پرس و جوهای آزمایشی

[1] Single Node/ Single Instance /Single Shard



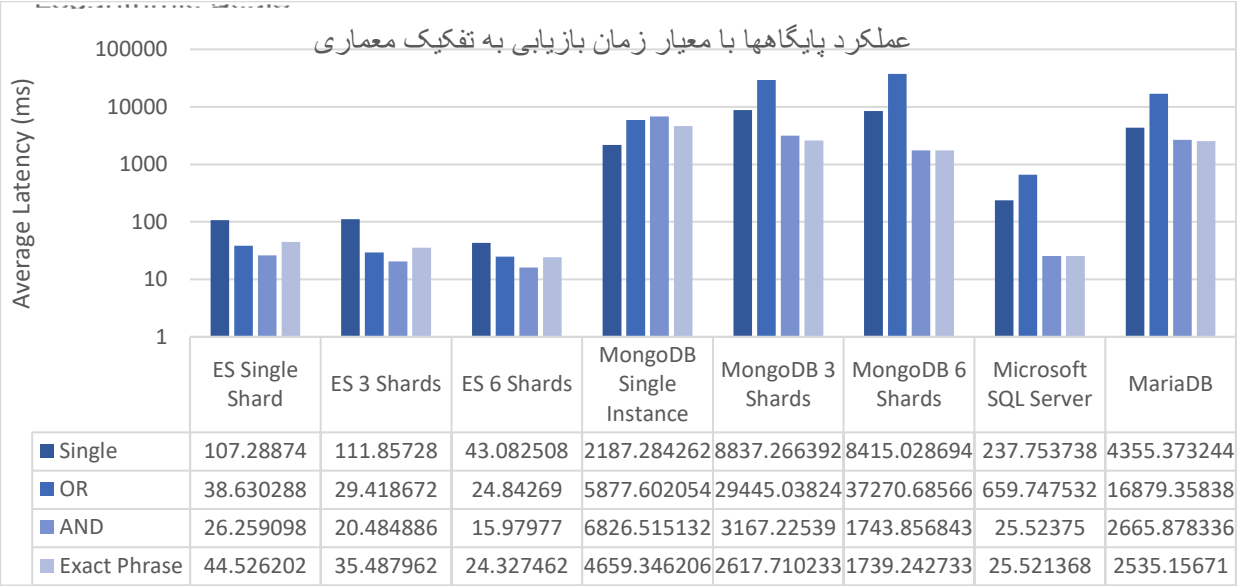
شکل۴-۲۳ مقایسه تاخیر زمانی پایگاه داده غیر رابطه ای ElasticSearch با در نظر گرفتن شاردینگ با چهار نوع پرس و جوی آزمایشی

شکل ۴-۲۲ به بررسی تاثیر افزایش تعداد Shard ها بر عملکرد جستجو و بازیابی تمام متن پایگاه داده MongoDB می پردازد. همانطور که قابل ملاحظه است با افزایش تعداد Shard ها عملکرد پرس و جوهای آزمایشی ANDو Exact Phrase بهبود پیدا می کند و در نقطه مقابل عملکرد پرس و جوهای آزمایشی Single وORاز نظر زمان بازیابی افت می کند.

اما با توجه به شکل ۴-۲۳ می توان دریافت که این افزایش تعداد Shard بطور کلی برای هر چهار نوع پرس و جوهای آزمایشی عملکرد پایگاه داده Elasticsearch را بهبود می دهد. بطور خلاصه می توان نتیجه گرفت افزایش تعداد shardها در MongoDBتاثیر منفی در کارایی و عملکرد خواهد گذاشت و این در حالیست که برای ElasticSearch افزایش shardها منجر به بهبود وضعیت عملکرد و کارآیی خواهد شد.

۴-۳-۲-۲ بررسی عملکرد پایگاههای داده رابطه ای و غیر رابطه ای در جستجو و بازیابی تمام متن به تفکیک معماری با معیار زمان بازیابی

در این قسمت به نمایش نمودارهای مقایسه ای کلی همه حالات معماری (پیاده سازی شده در آزمایشات)در یک نگاه می پردازیم تا بتوان به جمع بندی بهتری در خصوص انتخاب وضعیت بهینه پایگاه داده برگزیده دست یافت.



	ES Single Shard	ES 3 Shards	ES 6 Shards	MongoDB Single Instance	MongoDB 3 Shards	MongoDB 6 Shards	Microsoft SQL Server	MariaDB
Single	107.28874	111.85728	43.082508	2187.284262	8837.266392	8415.028694	237.753738	4355.373244
OR	38.630288	29.418672	24.84269	5877.602054	29445.03824	37270.68566	659.747532	16879.35838
AND	26.259098	20.484886	15.97977	6826.515132	3167.22539	1743.856843	25.52375	2665.878336
Exact Phrase	44.526202	35.487962	24.327462	4659.346206	2617.710233	1739.242733	25.521368	2535.15671

شکل ۴-۲۴ مقایسه زمان بازیابی پایگاههای داده رابطه ای و غیر رابطه ای در جستجوی تمام متن با چهار نوع پرس و جوهای آزمایشی به تفکیک معماری (لگاریتمی)

شکلهای ۴-۲۴ (که برای مشاهده بهتر به دو حالت ساده و لگاریتمی نیز نمایش داده شده است)، جمع بندی کلیه آزمایشات برای معیار زمان بازیابی است. مشاهده می شود SQL Server و ElasticSearch عملکرد بهتری را در زمان بازیابی جستجوی تمام متن در داده

های مرکز منطقه ای داشته اند. جمع بندی وضعیت پایگاههای داده در تمام حالات (انواع پرس و جوهای آزمایشی و شیوه های مختلف خوشه بندی[1] Sharded-Cluster) از نظرزمان بازیابی به شرح زیر قابل خلاصه شدن است.

Latency: ElasticSearch < SQL< MongoDB ~ MariaDB

در واقع همانطور که مشخص است، معیار زمان بازیابی در ElasticSearchدارای بهترین وضعیت (سریعترین زمان بازیابی با کمترین تاخیر) و سپس در رده دوم SQL و درنهایت MongoDB و MariaDB در جایگاه مشابهی در بدترین وضعیت با بیشترین زمان بازیابی (کندترین زمان بازیابی با بیشترین تاخیر) در جستجوی تمام متن قرار دارند

۴-۳-۲-۳ بررسی عملکرد پایگاههای داده رابطه ای و غیر رابطه ای در جستجوی تمام متن به تفکیک معماری با معیار کیفیت بازیابی داده

یکی از اهداف این پژوهش بررسی کیفیت داده های بازیابی شده می باشد؛ بدین منظور معیار کیفیت بازیابی داده ها QRRT[2] که تلفیقی از زمان بازیابی و کیفیت بازیابی داده است را به صورت زیر تعریف نمودیم؛ تا بتوان به جمع بندی بهتری در خصوص انتخاب کیفی پایگاه داده برگزیده دست یافت.

فرمول ۴-۱ $QRRT_{and,\ or} = QRR_{and,\ or}$ / (زمان بازیابی)

فرمول ۴-۲ $QRRT_{single,\ exact} = QRR_{single,\ exact}$ / (زمان بازیابی)

[1] Sharded Cluster
[2] Quality of Retrieved Result per Time

فرمول ۳-۴ $QRR^{1}_{or, and} = NAWC^{2}$ / (تعداد کل کلمات سند[3] بازیابی شده)

فرمول ۴-۴ $QRR_{single, exact}$ = (تعداد کلمات سند شده در مدرک) / (تعداد کل کلمات سند بازیابی شده)

NAWC میانگین نرمال شده تعداد کلمات بازیابی شده در نتیجه پرس و جوهای آزمایشی ANDیا OR می باشد. برای نرمال نمودن تعداد کلمات بر بیشینه تعداد بازیابی ها تقسیم میشود. مثال زیر چگونگی محاسبه NAWC را نشان می دهد:

Query = *receive* OR *proposed* OR *source*

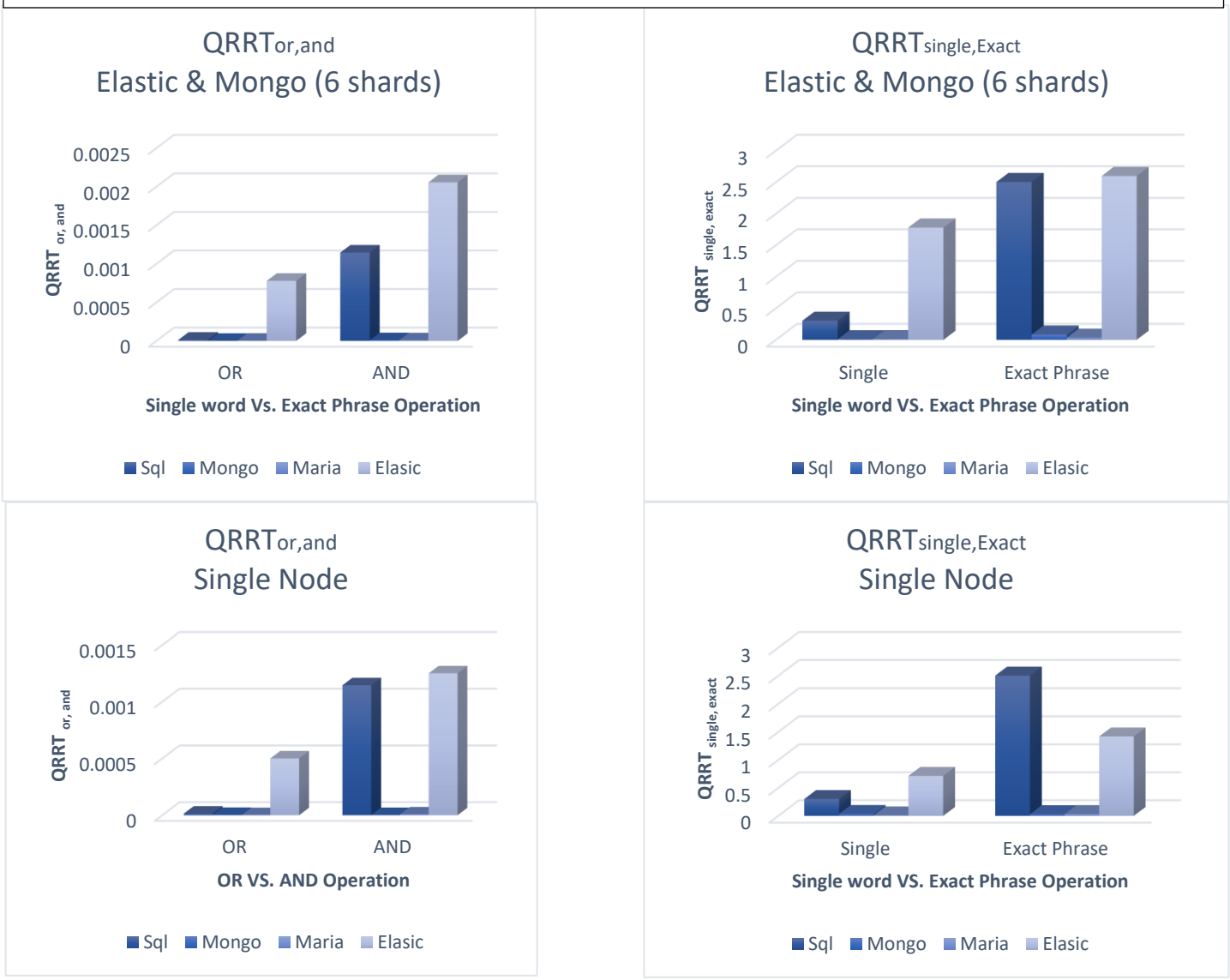
Number of each word in the best result:

	receive	*proposed*	*source*
Count	2	5	4

$NAWC = \left(\frac{2+4+5}{5}\right) / 3$

افزایش مقادیر عددی برای QRRT نشان دهنده عملکرد بهتر پایگاه‌های داده مورد بحث می باشد. در ادامه به نمایش نمودارهای مقایسه ای با معیار کیفیت بازیابی داده ها میپردازیم تا بتوان به جمع بندی بهتری در خصوص انتخاب وضعیت بهینه پایگاه داده برگزیده دست یافت.

[1] Quality of Retrieved Result
[2] Normalized Average of Word Count
[3] Document



شکل ۴-۲۵ مقایسه کیفیت بازیابی داده ها (QRRT)[1] در پایگاههای داده رابطه ای و غیر رابطه ای در جستجوی تمام متن

[1] QRRT: Quality of Retrieved Result (برای درک مفهوم " معیار کیفیت بازیابی داده"و سایر موارد مندرج در نمودار بالا به شرح توضیحات فرمولهای آن در بخش ۳-۲-۳-۴ مراجعه شود)

همانطور که در بخش های شکل ۴-۲۵ نمایش داده شده است، جمع بندی وضعیت پایگاههای داده در تمام حالات (انواع پرس و جوهای آزمایشی و شیوه های مختلف خوشه بندی Sharded-Cluster) از نظر معیار کیفیت بازیابی (QRRT) به شرح زیر قابل خلاصه شدن است.

QRRT: ElasticSearch > SQL> MongoDB ~ MariaDB

در واقع همانطور که مشخص است، معیار کیفیت بازیابی در ElasticSearchدارای بهترین وضعیت و سپس در رده دوم SQL و در رده بعدی MongoDB قرار دارد و نهایتا MariaDB در بدترین وضعیت از نظر معیار کیفیت بازیابی قرار دارد.

۴-۳-۳ جمع بندی نتایج جستجوی تمام متن

در مقیاسی که پایگاه های داده SQL Server، MariaDB، MongoDB و Elasticsearch را بوسیله جستجوی تمام متنمورد ارزیابی قرار دادیم، حکایت از برتری چشمگیر ElasticSearchو Microsoft SQL Server نسبت به دو پایگاه داده دیگر داشت. همچنین مشاهده شد که با افزایش تعداد Shardها در Elastic Search؛ Elasticsearch برتری خود را نسبت به Microsoft SQL Server افزایش می دهد.

فصل پنجم: بحث و نتیجه گیری

۵–۱ مقدمه

همانطور که گفته شد، امروزه داده ها به عنوان یک دارائی ملی شناخته می شوند . همچنین پردازش، تحلیل و استفاده از داده به عنوان یک عامل کلیدی برای رشد کلیه سازمانها تبدیل شده است و موجب مزیت رقابتی در کسب و کارها، محرک نوآوری، افزایش رقابت و اثرات مثبت اجتماعی خواهد شد. مرکز منطقه ای نیز از این قضیه مستثنی نمی باشد و با توجه به افزایش حجم داده ها بصورت تصاعدی (برای مثال در طی سال اخیر رشد ۵۰٪ داده های پایگاه داده آن را شاهد بر بستر پایگاه داده رابطه ای Microsoft SQL Server را شاهد بودیم) و با لحاظ نمودن ماموریتهای این مرکز در ارائه خدمات از طریق پایگاههای داده مقالات تمام متن و موتور جستجوی اختصاصی آن ، نیاز به ارائه راهکارهای نوین پردازشی یک ضرورت محسوب میشود. به بیان بهتر با توجه به چشم اندازهای آینده و رو به رشد مرکز منطقه ای و متناسب با سند راهبردی و همچنین اساسنامه آن استفاده از شیوه های نوین پردازش برای ارائه برنامه ها و خدمات اطلاع رسانی در منطقه تاکید شده است. با توجه به افزایش روزافزون حجم بزرگ داده ها، گسترش کیفی و کمی سرویسهای متعدد ارائه شده به سایر پژوهشگران در سطح ملی و بعضا بین المللی منطقه؛ حرکت به سمت ارتقا زیرساختهای نرم افزاری در قالب محاسبات توزیع شده و استفاده از روشهای نوین پردازش داده ها اهمیت می‌یابد.

لذا در این پژوهش بر آن شدیم تا برای کسب دانش کار با پایگاههای غیر رابطه ای NOSQL ، حرکت به سمت خوشه بندی (که از نقاط قوت پایگاههای داده غیر رابطه ای است)، و حل مشکلات

احتمالی آتی و پیش رو؛ با بکارگیری سیستمهای NOSQL به عنوان ابزار پردازش کلان داده ها در جستجو و بازیابی تمام متن، به مطالعه مقایسه ای وضعیت عملکردی وکارایی آنها در قیاس با پایگاه داده رابطه ای Microsoft SQL Serverمرکز منطقه ای بپردازیم.

فرضیه و سوالاتی که در این پژوهش مدنظر واقع شد و در طی این پژوهش با انجام آزمایشات در دو بخش اصلی توانستیم پاسخهای آن را دریابیم به شرح زیر است:

- *آیا با توجه به حجم داده ها و رشد آتی آنها پایگاههای غیر رابطه ای/سیستمهای NOSQL انتخاب مناسبی در راستای برآورده کردن نیازهای مرکز منطقه ای می باشد؟*

مسلما در آینده نزدیک در مرکز منطقه ای با حجم بی سابقه داده‌ها و نیازمندی‌های مقیاس-پذیری و کارایی برای این حجم از داده‌ها، این عرف را دستخوش تغییر می‌کند که پایگاه داده‌های رابطه‌ای تنها رویکرد مبتنی بر مدیریت داده‌ها هستند. بطور عمومی مبتنی بر تجارب دیگر پژوهشها و سازمانها با داده های حجیم، پایگاه داده‌های رابطه‌ای دارای مشکلاتی در مقیاس‌پذیری موثر، ذخیره سازی حجم بسیار بالا داده، پاسخ گویی کوییری های پیچیده در حجم بسیار بالای داده و ذخیره سازی داده‌های غیرساخت‌یافته می باشند.

لذا با توجه به کارایی مشهود در نتایج آزمایشات در فصل ۴ میتوان مدعی شد که پایگاههای داده غیر رابطه ای در حجم بالای داده باعث بهبود زمان و کیفیت بازیابی، توزیع خودکار داده و میزان تحمل خطای بالا برای ذخیره سازی داده های حجیم میشوند. لذا در آینده نزدیک استفاده از آنها برای مرکز ضروری است، چرا که این روزها با وجود چنین تکنولوژیهایی هزینه های مقیاس پذیری عمودی غیر قابل توجیه و بعضا تحمل ناپذیر بوده و باید با استفاده از تکنیک های جدید خوشه بندی در NOSQLپایگاههای داده غیررابطه ای، پایگاه داده ها به سمت معماریها و سیاستهای مقیاس پذیری افقی بروند.

- *کدامیک از پایگاههای غیر رابطه ای/سیستمهای NOSQL بهترین تناسب را با اهداف کاربردی مطرح شده در این طرح دارد؟*

همانطور که واضح است با گسترش روز افزون حجم داده ها و لذا نیاز به پایگاههای داده غیر رابطه ای؛ دانشگاهها و شرکتهای تجاری زیادی بصورت رایگان و غیر رایگان در حال توسعه بسترهای نرم افزاری برای پایگاههای داده غیر رابطه ای می‌باشند و تعداد زیادی از پلتفرمها، ابزارها و نرم افزارهای این حوزه در حال توسعه و تولید است. اما با توجه به نوع داده های حداکثری موجود و ذخیره شده در پایگاههای داده مرکز منطقه ای بهترین و محبوب ترین پایگاه داده غیر رابطه ای، پایگاه داده مبتنی بر سند (می باشد که قابلیت ذخیره سازی داده ها با فرمت هایی همچونJSON، BSON و XML را فراهم می آورند. این مدل برای ذخیره داده‌های بدون ساختار بسیار مناسب است چرا که هرسند می تواند از تعداد متفاوتی فیلد تشکیل شده باشد و لزومی ندارد که سند های مشابهی که در یک دسته بندی قرار می گیرند حتما از ساختار مشابهی پیروی کنند. این سیستمهای غیر رابطه ای قادرند پرس‌وجوهای تجمعی را اجرا کنند، نتایج را مرتب کنند و از ایندکس گذاری روی فیلدهای یک سند پشتیبانی کنند. نمونه‌هایی از این مدل پایگاه‌های داده غیر رابطه ای مبتنی بر سند Elastic, search MongoDB می باشند. که در پژوهش حاضر به صورت جامع مورد بحث و بررسی واقع شدند و نتایج آن در فصل ۴ ارائه گشت.

۵-۲ نتیجه گیری

در عصر اینترنت تعداد اسناد تمام متن روز به روز در حال افزایش است. در چنین شرایطی استخراج اطلاعات مورد نیاز از این اسناد با سرعت و کیفیت قابل قبول از اهمیت خاصی برخوردار است. راه حل پیاده سازی شده برای حل این مشکل در پایگاه‌های داده مختلف جستجو و بازیابی تمام متن می‌باشد .

هدف از این پژوهش بررسی عملکرد جستجو و بازیابی تمام متن برای پایگاه‌های داده Microsoft SQL Server و MariaDB در قیاس با MongoDB و Elasticsearch(از محبوب ترین پایگاه‌های داده غیر رابطه‌ای مبتنی بر سند جهانی) می‌باشد. با توجه به هدف پژوهش در گام اول به ارزیابی عملکرد پایگاه‌های داده معرفی شده به ازای عملیات اسکن و خوانش و به کمک معیار ارزیابی وای-سی-اس بی[1] پرداخته و در گام بعدی بطور دقیق تر عملکرد عملیات جستجو و بازیابی تمام متن در این پایگاه‌های داده بررسی شد. همچنین در این گام، در مورد تاثیر شاردینگ - که مرتبط با مهم ترین مزیت پایگاه‌های داده غیر رابطه‌ای نسبت به پایگاه‌های داده رابطه‌ای یعنی مقیاس پذیری افقی می‌باشد - بر عملکرد جستجو وبازیابی تمام متن MongoDB و Elasticsearch از دیدگاه معیار زمان بازیابی و کیفیت بازیابی بحث شد.

سایر نتایج حاصل شده به صورت جامع با جزییات در فصل ۴ شرح داده شد. در اینجا صرفا به اختصار به نتایج کلی و نهایی کار پس از انجام پژوهش اشاره می نماییم. در آزمایشات مربوط به جستجو و بازیابی تمام متن که عمده تمرکز این پژوهش محسوب میشود؛ در مقیاسی که پایگاه های داده SQL Server، MariaDB، MongoDB و Elasticsearch را بوسیله جستجوی تمام متن مورد

[1] YCSB Benchamrk

ارزیابی قرار دادیم، برتری چشمگیر ElasticSearchو Microsoft SQL Server نسبت به دو پایگاه داده دیگر مشهود بود. همچنین مشاهده شد که با افزایش تعداد Shardها در Elastic Search؛ Elasticsearch برتری خود را نسبت به Microsoft SQL Server افزایش می دهد.

بطور خلاصه میتوان اینگونه بیان نمود که؛ سیستمهای مربوط به NOSQL /پایگاه‌های داده غیررابطه ای MongoDB و رابطه ای MariaDB که عملکرد خوبی را در وای- سی –اس -بی ثبت کرده بودند، به عنوان پایگاه داده صرفا برای ذخیره سازی و پایداری داده ها مناسب هستند. این دو پایگاه داده در قسمت ارزیابی جستجوی تمام متن که عمده تمرکز این پژوهش محسوب می شود به مراتب عملکرد ضعیف تری نسبت به رقبای خود یعنی پایگاه‌های داده Microsoft SQL Server و Elasticsearchنشان دادند و بطور کلی در این قسمت بهترین عملکرد به Elasticsearch تعلق گرفت.

همچنین استفاده از قابلیت شاردینگ در پایگاه داده غیر رابطه ای بخصوص در حالت ElasticSearchباعث بهبود قابل توجه عملکرد آن در اجرای جستجوی تمام متن گشت. بنحوی که جمع بندی وضعیت پایگاههای داده در تمام حالات (انواع پرس و جوهای آزمایشی و شیوه های مختلف خوشه بندی[1] از نظرمعیار کیفیت بازیابی (QRRT) به شرح زیر قابل خلاصه شدن است.

QRRT: ElasticSearch > SQL> MongoDB ~ MariaDB

در واقع همانطور که مشخص است، معیار کیفیت بازیابی در ElasticSearchدارای بهترین وضعیت و سپس در رده دوم SQL و در رده بعدی MongoDB قرار دارد و نهایتا MariaDB در بدترین وضعیت از نظر معیار کیفیت بازیابی قرار دارد.

[1] Sharded-Cluster

با درنظر گرفتن نتایج این پژوهش با مطالعه انجام شده بر روی داده های مرکز منطقه ای و با تکیه بر ویژگی مقیاس پذیری افقی پایگاه‌های داده غیر رابطه ای، در زمان حاضر این پژوهش (با توجه به سرعت رشد تکنولوژی در حوزه سیستمهایNOSQL) در صورتیکه هدف حرکت به سمت استفاده از پایگاههای داده غیر رابطه ای باشد، بهترین تصمیم می تواند استفاده از پایگاه داده غیر رابطه ای قدرتمند همچون MongoDB به عنوان پایگاه داده اصلی و استفاده از Elasticsearch به عنوان ابزاری برای نمایه سازی[1] و جستجوی تمام متن بر روی پایگاه داده اصلی باشد. در صورتیکه همچنان نظر بر مقیاس پذیری عمودی و استفاده از پایگاههای داده رابطه ای در مرکز منطقه ای باشد، راهکار ترکیبی[2]استفاده از SQL به عنوان پایگاه داده و استفاده از موتور جستجوی قدرتمند ElasticSearch بر روی آن می باشد.

۵-۳ پیشنهادهای آینده

با درنظر گرفتن نتایج حاصل شده پژوهش، در پایان به عنوان **پیشنهادهای آینده** این طرح پژوهشی میتوان بر روی موارد زیر پژوهشهای بیشتری را با توجه به زیرساختهای فعلی و متناسب با ساختار و ماهیت داده های مرکز منطقه ای اطلاع رسانی علوم و فناوری لحاظ نمود:

- بررسی عمیق و جزیی بر نحوه عملکرد الگوریتمها و پرس و جوهای آزمایشی مختلف این دو پایگاه داده غیر رابطه ای MongoDB و ElasticSearch که خود از چالشهای مهم پایگاههای داده غیر رابطه ای /سیستمهای NOSQL در قیاس با پایگاههای داده رابطه ای محسوب میشود.

[1] Indexing
[2] Hybrid

- مطالعه و پژوهش بر چگونگی استفاده همزمان بصورت ترکیبی از هر دو بستر نرم افزاری پایگاههای داده رابطه ای و غیر رابطه ای و بررسی نتایج حاصل از آنها با هدف استفاده حداکثری از ویژگیهای هر دو (از جمله پایگاه داده رابطه ای SQL و پایگاه داده غیر رابطه ای یا به اصطلاح سیستم NOSQL- Elastic Search)
- مطالعه بر روی معماریهای متنوع احتمالی دیگر جهت برآورد هزینه سودمندی استفاده از مقیاس پذیری افقی در صورت فراهم آمدن زیرساختهای لازم

منابع و ماخذ

منابع و ماخذ فارسی

دولو، صدیقه و کوروش کیانی (۱۳۹۵). تحلیل تحرک فیزیکی افراد با استفاده از داده های کلان جمع آوری شده توسط برنامه کاربردی نصب شده روی موبایل افراد. اولین کنفرانس بین المللی دستاوردهای نوین پژوهشی در مهندسی برق و کامپیوتر۱۳۹۵، تهران، کنفدراسیون بین المللی مخترعان جهان(IFIA) ، دانشگاه جامع علمی کاربردی.https://www.civilica.com/Paper-CBCONF01-CBCONF01_1075.html

شعله ارسطوپور (۱۳۹۵). سند راهبردی مرکز منطقه ای اطلاع رسانی علوم و فناوری. شیراز: انتشارات مرکز منطقه ای اطلاع رسانی علوم و فناوری.

نوریان، فرشاد و سمانه حجازی (۱۳۹۲). کاربرد کلان-داده ها در نقد توسعه مبتنی بر حمل ونقل عمومی. , فصلنامه مطالعات شهری دانشگاه کردستان دوره ۸،شماره ۸ : صفحه ۸۳-۹۱.

مدبرفرزام، نیلوفر و احمد فراهی (۱۳۹۴). مدیریت داده های حجیم با استفاده از سیستم های NoSQL، کنفرانس بین المللی پژوهش های کاربردی در فناوری اطلاعات ۱۳۹٤، کامپیوتر و مخابرات، تربت حیدریه، شرکت مخابرات خراسان رضوی، https://www.civilica.com/Paper-ITCC01-ITCC01_124.html

نصری فلاح، پروین (۱۳۹۵). کاربرد و چالش های کلان داده ها، اولین کنفرانس بین المللی چشم انداز های نو در مهندسی برق و کامپیوتر۱۳۹۵، تهران، کنفدراسیون بین المللی مخترعان جهان(IFIA) ، دانشگاه جامع علمی کاربردی، https://www.civilica.com/Paper-NPECE01-NPECE01_388.html

منابع و ماخذ لاتین

Abramova, V., & Bernardino, J. (2013). NoSQL databases: MongoDB vs Cassandra. In Proceedings of the international C* conference on computer science and software engineering (pp. 14-22).

Abramova, V., J. Bernardino, P. Furtado, Which nosql database? a performance overview, Open J. Databases. 1 (2014) 17–24.

Amazon DynamoDB: Fast and flexible NoSQL database service for any scale.(2017). Available from: https://aws.amazon.com/dynamodb/. Accessed September 20, 2017.

Amazon SimpleDB (2017). Available from: https://aws.amazon.com/simpledb (2017). Accessed September 2017.

Carpenter, J., & Hewitt, E. (2020). Cassandra: the definitive guide: distributed data at web scale. O'Reilly Media.

Chambers, B., & Zaharia, M. (2018). Spark: The definitive guide: Big data processing made simple. "O'Reilly Media, Inc.".

Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., ... & Gruber, R. E. (2008). Bigtable: A distributed storage system for structured data. ACM Transactions on Computer Systems (TOCS), 26(2), 1-26.

Dede, E., Govindaraju, M., Gunter, D., Canon, R. S., & Ramakrishnan, L. (2013). Performance evaluation of a mongodb and hadoop platform for scientific data analysis. In Proceedings of the 4th ACM workshop on Scientific cloud computing (pp. 13-20).

Chen, J. K., & Lee, W. Z. (2019). An Introduction of NoSQL Databases Based on Their Categories and Application Industries. Algorithms, 12(5), 106.

Cooper B.F., A. Silberstein, E. Tam, R. Ramakrishnan, R. Sears, Benchmarking cloud serving systems with YCSB, Proc. 1st ACM Symp. Cloud Comput. - SoCC '10. (2010) 143–154. doi:10.1145/1807128.1807152.

DB-Engines Ranking (2017). Available from: https://db-engines.com/en/ranking. Accessed September 20, 2017.

Divya, M. S., & Goyal, S. K. (2013). ElasticSearch: An advanced and quick search technique to handle voluminous data. Compusoft, 2(6), 171.

Dixit, B. (2016). Elasticsearch Essentials. Packt Publishing Ltd. ElasticSearch. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/5.5/modules-cluster.html. Accessed August 2, 2019.

Elasticsearch Resiliency Status (2019). Available from: https://www.elastic.co/guide/en/elasticsearch/resiliency/current/index.html Accessed August 2, 2019.

ElasticSearch Manual Document1 (2019). Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-node.html . Accessed August 2, 2019.

ElasticSearch Manual Document2 (2019). Available from: Document1https://www.elastic.co/guide/en/elasticsearch/reference/5.5/important-settings.html.Accessed August 2, 2019.

Ghazal, A.,T. Rabl, M. Hu, F. Raab, M. Poess, A. Crolotte, H.-A. Jacobsen, BigBench: towards an industry standard benchmark for big data analytics, in: Proc. 2013 ACM SIGMOD Int. Conf. Manag. Data, 2013: pp. 1197–1208.

Github. Avaialble from: https://github.com/elastic/elasticsearch/issues/10933. Accessed August 2, 2019.

Google Trends. "Using NOSQL versus SQL", Available from: https://trends.google.com/. Accessed August 2, 2019.

Gudivada, V. N., Rao, D., & Raghavan, V. V. (2014). NoSQL systems for big data management. In 2014 IEEE World congress on services (pp. 190-197). IEEE.

Hadoop (2019). Available from : http://hadoop.apache.org/. Accessed September 20, 2019.

Han, J., Haihong, E., Le, G., & Du, J. (2011). Survey on NoSQL database. In 2011 6th international conference on pervasive computing and applications (pp. 363-366). IEEE.

HBase (2017). Available from: https://hbase.apache.org/. Accessed September 20, 2017.

Cloud Serv. Comput., IEEE, 2011: pp. 336–341. doi:10.1109/CSC.2011.6138544.

Hoberman, S. (2014). Data Modeling for MongoDB: Building Well-Designed and Supportable MongoDB Databases. Technics Publications.

Hypertable. Available from: www.hypertable.com. Accessed September 20, 2017.

Kenler, E., & Razzoli, F. (2015). MariaDB Essentials. Packt Publishing Ltd.

Li, Y., & Manoharan, S. (2013). A performance comparison of SQL and NoSQL databases. In 2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM) (pp. 15-19). IEEE.

Macedo, T., & Oliveira, F. (2011). Redis Cookbook: Practical Techniques for Fast Data Manipulation. "O'Reilly Media, Inc."

Mavridis, I., H. Karatza, Performance evaluation of cloud-based log file analysis with Apache Hadoop and Apache Spark, J. Syst. Softw. 125 (2017) 133–151.

MongoDB (2019). Available from: https://www.mongodb.com/industries/retail. Accessed August 2, 2019.

MongoDb Manual Document (2019). Available from: https://docs.mongodb.com/manual/tutorial/deploy-shard-cluster/. Accessed August 2, 2019.

Oracle-RDBMS (2017). Available from: https://www.oracle.com/database/what-is-a-relational-database. Accessed September 20, 2017.

Patil, M. M., Hanni, A., Tejeshwar, C. H., & Patil, P. (2017). A qualitative analysis of the performance of MongoDB vs MySQL database based on insertion and retriewal operations using a web/android application to explore load balancing—Sharding in MongoDB and its advantages. In 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) (pp. 325-330). IEEE.

Riak (2017). Available from: http://basho.com/products/#riak. Accessed September 20, 2017.

Redis (2017). Available from: https://redis.io/. Accessed September 20, 2017.

Vaish, G. (2013). Getting started with NoSQL. Packt Publishing Ltd.

Voldmort Project (2017). Available from: http://project-voldemort.com. Accessed August 2, 2019.

Yahoo! Cloud Serving Benchmark (YCSB) (2019). Avaialable from: https://github.com/brianfrankcooper/YCSB/wiki. Accessed August 2, 2019.