تحلیل جامع عملکرد ArangoDB: مطالعه مقایسه‌ای پایگاه‌داده‌های NoSQL چندمدلی

علی بخشا
دانشکده علوم داده
دانشگاه خلیج فارس
alibakhsha@mehr.pgu.ac.ir

۱۴ آذر ۱۴۰۴

چکیده

گسترش کلان‌داده‌ها و نیازمندی‌های متنوع کاربردی، تکامل پایگاه‌داده‌های NoSQL را به همراه داشته است، به طوری که پایگاه‌داده‌های چندمدلی (Multi-model) به عنوان یک راهکار یکپارچه برای مدیریت انواع داده‌های ناهمگن ظهور کرده‌اند. ArangoDB به عنوان یک پایگاه‌داده چندمدلی بومی که از مدل‌های داده‌ای سند (Document)، گراف (Graph) و کلید-مقدار (Key-Value) پشتیبانی می‌کند، جایگزینی جذاب برای استقرار چندین پایگاه‌داده تخصصی ارائه می‌دهد. این مقاله به تحلیل جامع عملکرد ArangoDB می‌پردازد و آن را با پایگاه‌داده‌های پیشرو NoSQL از جمله MongoDB (ذخیره‌ساز سند)، Neo4j (پایگاه‌داده گراف)، Redis (ذخیره‌ساز کلید-مقدار) و Apache Cassandra (ذخیره‌ساز ستونی) مقایسه می‌کند. از طریق بنچمارک‌گیری سیستماتیک در بارهای کاری مختلف، ما عملکرد پرس‌وجو، توان عملیاتی، تاخیر، مقیاس‌پذیری و کارایی منابع را ارزیابی می‌کنیم. نتایج تجربی ما نشان می‌دهد که ArangoDB در هر سه مدل داده‌ای عملکرد رقابتی دارد و در عین حال انعطاف‌پذیری برتری برای بارهای کاری ترکیبی که نیاز به پرس‌وجوهای چندمدلی دارند، ارائه می‌دهد. به طور خاص، ArangoDB برای پیمایش‌های پیچیده گراف در گراف‌های با اندازه متوسط (تا 10 میلیون یال)، 15 تا 20 درصد تاخیر کمتری نسبت به پایگاه‌داده‌های تخصصی گراف نشان می‌دهد و 85 تا 95 درصد از توان عملیاتی MongoDB را برای عملیات سند حفظ می‌کند. این مطالعه بینش‌هایی را در مورد موازنه بین رویکردهای تخصصی و چندمدلی ارائه می‌دهد و راهنمایی‌هایی برای انتخاب پایگاه‌داده بر اساس نیازمندی‌های کاربردی فراهم می‌کند.

واژگان کلیدی: ArangoDB، پایگاه‌داده‌های NoSQL، پایگاه‌داده‌های چندمدلی، بنچمارک عملکرد، پایگاه‌داده‌های گراف، ذخیره‌سازهای سند، مقایسه پایگاه‌داده.

۱ مقدمه

چشم‌انداز سیستم‌های مدیریت پایگاه‌داده در دو دهه گذشته دستخوش تحولات قابل توجهی شده است. پایگاه‌داده‌های رابطه‌ای سنتی، اگرچه قدرتمند و بالغ هستند، در پاسخگویی به نیازهای مقیاس‌پذیری، انعطاف‌پذیری و عملکرد برنامه‌های مدرن داده‌محور با چالش‌هایی روبرو هستند [۱]. ظهور پایگاه‌داده‌های NoSQL با قربانی کردن ویژگی‌های سخت‌گیرانه ACID و اسکیمای رابطه‌ای به نفع مقیاس‌پذیری افقی، مدل‌های داده‌ای منعطف و عملکرد بهینه برای موارد استفاده خاص، به این محدودیت‌ها پاسخ داد [۲].

۱.۱ تکامل NoSQL

پایگاه‌داده‌های NoSQL معمولاً به چهار دسته اصلی تقسیم می‌شوند: ذخیره‌سازهای سند (مانند MongoDB, CouchDB)، ذخیره‌سازهای کلید-مقدار (مانند Redis, DynamoDB)، ذخیره‌سازهای خانواده ستونی (مانند Cassandra, HBase) و پایگاه‌داده‌های گراف (مانند Neo4j, JanusGraph) [۳]. هر دسته در سناریوهای خاصی برتری دارد: ذخیره‌سازهای سند انعطاف‌پذیری را برای داده‌های نیمه‌ساختاریافته فراهم می‌کنند، ذخیره‌سازهای کلید-مقدار عملکرد خواندن/نوشتن فوق‌العاده‌ای برای پرس‌وجوهای ساده ارائه می‌دهند، ذخیره‌سازهای ستونی تحلیل‌های کارآمد روی مجموعه داده‌های بزرگ را ممکن می‌سازند و پایگاه‌داده‌های گراف پرس‌وجوهای مبتنی بر روابط را بهینه می‌کنند.

با این حال، برنامه‌های کاربردی مدرن به طور فزاینده‌ای به چندین مدل داده به طور همزمان نیاز دارند. برای مثال، یک پلتفرم رسانه اجتماعی ممکن است به ذخیره‌سازی سند برای پروفایل کاربران، ساختارهای گراف برای شبکه‌های اجتماعی و دسترسی کلید-مقدار برای مدیریت نشست‌ها (Session) نیاز داشته باشد. به طور سنتی، این امر مستلزم استقرار و نگهداری چندین سیستم پایگاه‌داده است که پیچیدگی عملیاتی، چالش‌های همگام‌سازی داده‌ها و هزینه‌های زیرساخت را افزایش می‌دهد [۴].

۲.۱ پارادایم پایگاه‌داده چندمدلی

پایگاه‌داده‌های چندمدلی برای رفع این پراکندگی با پشتیبانی از چندین مدل داده در یک موتور پایگاه‌داده واحد ظهور کردند. ArangoDB که در سال 2011 معرفی شد، یک رویکرد چندمدلی بومی (Native) را نشان می‌دهد که از پایه برای مدیریت اسناد، گراف‌ها و جفت‌های کلید-مقدار با یک زبان پرس‌وجوی واحد طراحی شده است [۵]. این در تضاد با پایگاه‌داده‌هایی است که پشتیبانی چندمدلی را به صورت گذشته‌نگر از طریق افزونه‌ها اضافه کرده‌اند.

مزایای نظری پایگاه‌داده‌های چندمدلی عبارتند از:

- کاهش پیچیدگی عملیاتی: یک سیستم پایگاه‌داده واحد برای استقرار، نظارت و نگهداری.
- سازگاری داده‌ها: معناشناسی تراکنش یکپارچه در سراسر مدل‌های داده.

- انعطاف‌پذیری پرس‌وجو: توانایی ترکیب عملیات در مدل‌های مختلف.
- کارایی هزینه: کاهش سربار زیرساخت و مجوزها.

۳.۱ انگیزه و اهداف تحقیق

با وجود این مزایای نظری، سوالات اساسی در مورد پیامدهای عملکردی عملی رویکرد چندمدلی باقی مانده است. آیا تعمیم‌یافتگی ذاتی در پشتیبانی از چندین مدل، عملکرد را در مقایسه با پایگاه‌داده‌های تخصصی به خطر می‌اندازد؟ آیا یک زبان پرس‌وجوی واحد می‌تواند عملیات را در پارادایم‌های مختلف داده به طور کارآمد بیان کند؟ عملکرد با افزایش حجم داده‌ها چگونه مقیاس می‌شود؟

این مطالعه با هدف پاسخگویی به این سوالات از طریق بنچمارک‌گیری و تحلیل جامع انجام شده است. اهداف خاص تحقیق ما عبارتند از:

۱. توصیف عملکرد: اندازه‌گیری سیستماتیک عملکرد ArangoDB در بارهای کاری سند، گراف و کلید-مقدار.

۲. تحلیل مقایسه‌ای: بنچمارک ArangoDB در برابر پایگاه‌داده‌های تخصصی پیشرو در هر دسته.

۳. ارزیابی مقیاس‌پذیری: ارزیابی ویژگی‌های مقیاس‌پذیری افقی و عملکرد خوشه‌ای.

۴. مزیت چندمدلی: کمی‌سازی مزایای عملکرد برای پرس‌وجوهایی که چندین مدل داده را در بر می‌گیرند.

۵. تحلیل موازنه: شناسایی سناریوهایی که پایگاه‌داده‌های تخصصی مزایای خود را حفظ می‌کنند.

۴.۱ مشارکت‌ها

این مقاله مشارکت‌های زیر را ارائه می‌دهد:

- بنچمارک‌های جامع عملکرد که ArangoDB را با MongoDB، Neo4j، Redis و Cassandra در بارهای کاری استاندارد مقایسه می‌کند.
- تحلیل عملکرد پرس‌وجو برای عملیات سند، پیمایش گراف و الگوهای دسترسی کلید-مقدار.
- ارزیابی قابلیت‌های پرس‌وجوی چندمدلی منحصر به فرد برای ArangoDB.
- تحلیل مقیاس‌پذیری در پیکربندی‌های خوشه‌ای از تک‌گره تا استقرارهای توزیع‌شده.
- توصیه‌های عملی برای انتخاب پایگاه‌داده بر اساس ویژگی‌های بار کاری.

۵.۱ سازماندهی مقاله

ادامه این مقاله به شرح زیر سازماندهی شده است: بخش دوم کارهای مرتبط در ارزیابی عملکرد NoSQL و پایگاه‌داده‌های چندمدلی را مرور می‌کند. بخش سوم معماری و ویژگی‌های کلیدی ArangoDB را توصیف می‌کند. بخش چهارم روش‌شناسی تجربی و طراحی بنچمارک ما را شرح می‌دهد. بخش پنجم نتایج عملکرد را در انواع مختلف بار کاری ارائه می‌دهد. بخش ششم موارد استفاده و پیامدهای عملی را مورد بحث قرار می‌دهد. بخش هفتم یافته‌ها را تحلیل و موازنه‌ها را شناسایی می‌کند. بخش هشتم با توصیه‌ها و جهت‌گیری‌های تحقیقاتی آینده نتیجه‌گیری می‌کند.

۲ مرور ادبیات و کارهای مرتبط

۱.۲ طبقه‌بندی پایگاه‌داده‌های NoSQL

اصطلاح «NoSQL» شامل معماری‌های متنوع پایگاه‌داده است که با جدایی از مدل‌های رابطه‌ای سنتی متحد شده‌اند. طبقه‌بندی تاثیرگذار Cattell [۶] دسته‌های بنیادی را بر اساس مدل‌های داده و تضمین‌های سازگاری ایجاد کرد. کار بعدی توسط Han و همکاران [۳] این طبقه‌بندی را اصلاح کرد و بر پیامدهای قضیه CAP برای انواع مختلف NoSQL تاکید نمود.

بررسی‌های اخیر [۲، ۷] بلوغ اکوسیستم‌های NoSQL و پذیرش فزاینده آن‌ها در محیط‌های تولیدی را برجسته می‌کنند. ذخیره‌سازهای سند مانند MongoDB به دلیل اسکیمای منعطف خود برای مدیریت محتوا و بک‌اند موبایل محبوبیت پیدا کرده‌اند [۸]. پایگاه‌داده‌های گراف مانند Neo4j برای شبکه‌های اجتماعی، موتورهای توصیه و کشف تقلب که در آن‌ها پرس‌وجوهای مبتنی بر روابط غالب هستند، استاندارد شده‌اند [۹].

۲.۲ بنچمارک عملکرد NoSQL

ارزیابی عملکرد پایگاه‌داده‌های NoSQL توجه قابل توجهی را به خود جلب کرده است. بنچمارک سرویس‌دهی ابری Yahoo! (YCSB) [۱۰] الگوهای بار کاری استانداردی را برای مقایسه ذخیره‌سازهای کلید-مقدار و سند ایجاد کرد. افزونه‌هایی مانند YCSB++ [۱۱] پشتیبانی از پایگاه‌داده‌های گراف و الگوهای پرس‌وجوی غنی‌تر را اضافه کردند.

مطالعات مقایسه‌ای موازنه‌های عملکرد را در دسته‌های NoSQL بررسی کرده‌اند. Li و Manoharan [۱۲] پایگاه‌داده‌های MongoDB، Cassandra و HBase را مقایسه کردند و تغییرات عملکرد قابل توجهی را بر اساس نسبت‌های خواندن/نوشتن و اندازه داده‌ها یافتند. Jouili و Vansteenberghe [۱۳] پایگاه‌داده‌های گراف را بنچمارک کردند و نشان دادند که پیچیدگی اسکیما و عمق پیمایش به طور قابل توجهی بر تاخیر پرس‌وجو تاثیر می‌گذارد.

تحقیقات در مورد عملکرد ذخیره‌ساز سند به ویژه بر MongoDB به دلیل محبوبیت آن متمرکز بوده است. مطالعات اعتبارسنجی [۱۴] کارایی MongoDB را برای بازیابی اسناد نشان

می‌دهند اما کاهش عملکرد را برای پرس‌وجوهای تجمیعی پیچیده در مجموعه‌های بزرگ شناسایی می‌کنند. این یافته‌ها انگیزه ما برای گنجاندن هر دو الگوی پرس‌وجوی ساده و پیچیده در مجموعه بنچمارک ما بود.

۳.۲ تحقیق در پایگاه‌داده‌های چندمدلی

پایگاه‌داده‌های چندمدلی یک حوزه تحقیقاتی نسبتاً جدید را نشان می‌دهند. Lu و همکاران [۴] یک بررسی جامع ارائه می‌دهند که بین سیستم‌های چندمدلی بومی (طراحی شده برای چندین مدل از ابتدا) و سیستم‌های بازسازی‌شده (پایگاه‌داده‌های تک‌مدلی که برای پشتیبانی از مدل‌های اضافی توسعه یافته‌اند) تمایز قائل می‌شوند.

پیامدهای عملکردی معماری‌های چندمدلی تحقیقات تجربی محدودی دریافت کرده‌اند. Ghazal و همکاران [۱۵] بنچمارک‌هایی را برای تحلیل‌های چندمدلی پیشنهاد کردند اما عمدتاً بر پلتفرم‌های کلان‌داده متمرکز بودند تا پایگاه‌داده‌های عملیاتی. Wingerath و همکاران [۱۶] الگوهای ماندگاری چندزبانه را بررسی کردند و سربار هماهنگی را به عنوان یک چالش کلیدی در استفاده از چندین پایگاه‌داده تخصصی شناسایی کردند.

تحقیقات خاص ArangoDB در ادبیات آکادمیک پراکنده است. مستندات و مقاله‌های سفید از ArangoDB GmbH [۵] تصمیمات معماری را توصیف می‌کنند اما فاقد اعتبارسنجی مستقل عملکرد هستند. بنچمارک‌های اجتماعی وجود دارند اما اغلب فاقد دقت روش‌شناختی یا معیارهای مقایسه‌ای هستند.

۴.۲ عملکرد پایگاه‌داده گراف

عملکرد پایگاه‌داده گراف به دلیل ماهیت عملیات پیمایش ویژگی‌های منحصر به فردی دارد. Holzschuher و Peinl [۱۷] نشان دادند که پایگاه‌داده‌های گراف بومی مانند Neo4j برای پیمایش‌های عمیق به طور قابل توجهی از پایگاه‌داده‌های رابطه‌ای بهتر عمل می‌کنند اما برای پرس‌وجوهای کم‌عمق همگرایی عملکردی را نشان دادند.

کارهای اخیر [۱۸] روی زبان‌های پرس‌وجوی گراف و تکنیک‌های بهینه‌سازی، وضعیت هنر را پیش برده است. با این حال، اکثر مطالعات بر پایگاه‌داده‌های گراف تخصصی تمرکز دارند تا سیستم‌های چندمدلی که از عملیات گراف پشتیبانی می‌کنند.

۵.۲ شکاف تحقیقاتی

در حالی که ادبیات موجود به طور گسترده پایگاه‌داده‌های NoSQL تخصصی را پوشش می‌دهد، تحلیل جامع عملکرد پایگاه‌داده‌های چندمدلی بومی همچنان محدود است. سوالاتی در مورد جریمه (یا سود) عملکردی پشتیبانی چندمدلی، به ویژه برای بارهای کاری که چندین مدل داده را ترکیب می‌کنند، فاقد پاسخ‌های تجربی است. این مطالعه این شکاف را از طریق بنچمارک‌گیری سیستماتیک ArangoDB در برابر پایگاه‌داده‌های تخصصی پیشرو در رده خود برطرف می‌کند.

۳ معماری و ویژگی‌های ArangoDB

۱.۳ بنیاد چندمدلی

ArangoDB یک معماری چندمدلی بومی را پیاده‌سازی می‌کند که در آن اسناد، گراف‌ها و جفت‌های کلید-مقدار شهروندان درجه یک هستند نه افزودنی‌هایی به یک مدل هسته‌ای. این فلسفه طراحی بر هر لایه از سیستم، از موتور ذخیره‌سازی تا بهینه‌سازی پرس‌وجو، تاثیر می‌گذارد.

۱.۱.۳ مدل داده یکپارچه

در هسته خود، ArangoDB تمام داده‌ها را به عنوان اسناد JSON ذخیره می‌کند. یال‌های گراف به عنوان اسنادی با ویژگی‌های خاص `_from` و `_to` که به اسناد راس ارجاع می‌دهند، نمایش داده می‌شوند. دسترسی کلید-مقدار از طریق جستجوی مستقیم سند توسط کلید اصلی حاصل می‌شود. این مدل ذخیره‌سازی یکپارچه امکانات زیر را فراهم می‌کند:

- انعطاف‌پذیری نوع: رئوس و یال‌ها می‌توانند ویژگی‌های دلخواه داشته باشند.
- یکپارچگی پرس‌وجو: پیوندهای (Join) بدون درز بین مجموعه‌هایی از انواع مختلف.
- ایندکس‌گذاری سازگار: انواع ایندکس یکسان قابل اجرا در همه مدل‌ها.

۲.۱.۳ انواع مجموعه (**Collection**)

ArangoDB دو نوع مجموعه تعریف می‌کند:

۱. مجموعه‌های سند: مجموعه‌های استاندارد ذخیره‌کننده اسناد JSON، مشابه مجموعه‌های MongoDB.

۲. مجموعه‌های یال: مجموعه‌های تخصصی ذخیره‌کننده یال‌های گراف با ارجاعات اجباری `_from` و `_to`.

۲.۳ زبان پرس‌وجوی ArangoDB (AQL)

AQL یک نحو شبه SQL اعلانی برای همه مدل‌های داده فراهم می‌کند. ویژگی‌های کلیدی عبارتند از:

۱.۲.۳ پرس‌وجوهای سند

```
FOR user IN users
  FILTER user.age > 25
  AND user.country == "USA"
  SORT user.name
  LIMIT 100
  RETURN user
```

Listing 1: AQL Document Query Example

۲.۲.۳ پیمایش‌های گراف

```
FOR v, e, p IN 1..3 OUTBOUND
  'users/john' GRAPH 'social'
  FILTER v.active == true
  RETURN {
    person: v.name,
    path_length: LENGTH(p.vertices)
  }
```

Listing 2: AQL Graph Traversal Example

۳.۲.۳ پرس‌وجوهای چندمدلی

مزیت متمایز AQL در پرس‌وجوهایی که مدل‌ها را ترکیب می‌کنند، ظاهر می‌شود:

```
FOR user IN users
  FOR product IN 1..1 OUTBOUND
    user purchases
  FILTER product.price > 100
  COLLECT category = product.category
  AGGREGATE total = SUM(product.price)
  RETURN {category, total}
```

Listing 3: Multi-model Query Example

این پرس‌وجو فیلترینگ سند، پیمایش گراف و تجمیع را در یک دستور واحد ترکیب می‌کند – عملیاتی که در معماری چندزبانه به چندین پرس‌وجو در پایگاه‌داده‌های مختلف نیاز دارد.

۳.۳ موتور ذخیره‌سازی

ArangoDB از RocksDB به عنوان موتور ذخیره‌سازی پیش‌فرض خود استفاده می‌کند (جایگزین موتور قبلی MMFiles). RocksDB موارد زیر را ارائه می‌دهد:

- معماری درخت ادغام ساختاریافته با لاگ (**LSM**): بهینه شده برای بارهای کاری سنگین نوشتن.
- فشرده‌سازی: کاهش ردپای ذخیره‌سازی.
- سازگاری قابل تنظیم: سیاست‌های همگام‌سازی قابل پیکربندی که دوام و عملکرد را متعادل می‌کنند.

۴.۳ مکانیسم‌های ایندکس‌گذاری

ArangoDB از چندین نوع ایندکس پشتیبانی می‌کند که در تمام انواع مجموعه قابل استفاده هستند:

جدول ۱: انواع ایندکس ArangoDB و موارد استفاده

نوع ایندکس	بهترین برای	پیچیدگی
Persistent	پرس‌وجوهای عمومی	$O(\log n)$
Hash	جستجوهای تساوی	$O(1)$
Skiplist	پرس‌وجوهای دامنه‌ای	$O(\log n)$
Fulltext	جستجوی متن	$O(k)$
Geo	پرس‌وجوهای مکانی	$O(\log n)$
TTL	اسناد منقضی‌شونده	$O(1)$

مجموعه‌های یال به طور خودکار ایندکس‌های Hash را روی ویژگی‌های `_from` و `_to` ایجاد می‌کنند که پیمایش‌های گراف را تسریع می‌کند.

۵.۳ خوشه‌بندی و مقیاس‌پذیری

معماری خوشه ArangoDB از طراحی بدون ارباب (Masterless) با سه نوع گره استفاده می‌کند:

۱. عامل‌ها (**Agents**): نگهداری پیکربندی خوشه با استفاده از اجماع Raft.

۲. هماهنگ‌کننده‌ها (**Coordinators**): مدیریت درخواست‌های کلاینت و هماهنگی پرس‌وجو.

۳. سرورهای پایگاه‌داده (**DB-Servers**): ذخیره و مدیریت شاردهای داده.

۱.۵.۳ توزیع داده

مجموعه‌ها به طور خودکار با استفاده از درهم‌سازی سازگار (Consistent Hashing) در سرورهای DB شارد می‌شوند. فاکتور تکرار و تعداد شاردها برای هر مجموعه قابل پیکربندی است که امکان بهینه‌سازی دقیق عملکرد را فراهم می‌کند.
برای مجموعه‌های گراف، ArangoDB دو استراتژی شاردینگ ارائه می‌دهد:

- شاردینگ تصادفی: توزیع استاندارد مبتنی بر هش.
- گراف‌های هوشمند (**Smart Graphs**): هم‌مکان کردن رئوس و یال‌های مرتبط برای به حداقل رساندن پرش‌های شبکه در طول پیمایش‌ها.

۶.۳ پشتیبانی از تراکنش

ArangoDB تراکنش‌های ACID را با ویژگی‌های زیر ارائه می‌دهد:

- تراکنش‌های تک‌سند: همیشه مطابق با ACID.
- تراکنش‌های چندسند: ACID در یک مجموعه واحد.
- تراکنش‌های چندمجموعه: ACID در استقرارهای تک‌سرور؛ سازگاری نهایی (Eventual Consistency) در خوشه‌ها (با استثنائاتی برای گراف‌های هوشمند).

این مدل تراکنش نشان‌دهنده موازنه‌ای بین تضمین‌های سازگاری و مقیاس‌پذیری خوشه است که ArangoDB را بین سیستم‌های کاملاً سازگار (RDBMS سنتی) و پایگاه‌داده‌های NoSQL با سازگاری نهایی قرار می‌دهد.

۷.۳ بهینه‌سازی پرس‌وجو

بهینه‌ساز AQL از استراتژی‌های متعددی استفاده می‌کند:

۱. انتخاب ایندکس: انتخاب خودکار ایندکس‌های بهینه.
۲. بهینه‌سازی الحاق: مرتب‌سازی مجدد الحاق‌ها برای کارایی.
۳. هل دادن فیلتر به پایین: انتقال شروط (Predicates) به نزدیکی دسترسی داده.
۴. اجرای موازی: توزیع اجرای پرس‌وجو در شاردهای مختلف.

پروفایلینگ پرس‌وجو برنامه اجرا را نشان می‌دهد و توسعه‌دهندگان را قادر می‌سازد گلوگاه‌های عملکرد را شناسایی کنند:

```
FOR user IN users
  FILTER user.status == "active"
  RETURN user
/* Execution plan uses
   persistent index on status */
```

Listing 4: Query Profiling Example

۴ روش‌شناسی تجربی

۱.۴ اصول طراحی بنچمارک

مجموعه بنچمارک ما از اصول زیر پیروی می‌کند:

۱. تنوع بار کاری: پوشش سناریوهای سنگین-خواندن، سنگین-نوشتن و متعادل.

۲. واقع‌گرایی داده‌ها: استفاده از توزیع‌های داده و الگوهای پرس‌وجوی واقعی.

۳. انصاف: اطمینان از بهینه‌سازی قابل مقایسه در تمام پایگاه‌داده‌ها.

۴. تکرارپذیری: مستندسازی تمام پارامترهای پیکربندی و روش‌های تست.

۵. اعتبار آماری: اجراهای متعدد با بازه‌های اطمینان.

۲.۴ انتخاب پایگاه‌داده و نسخه‌ها

ما پایگاه‌داده‌های زیر را برای مقایسه انتخاب کردیم:

جدول ۲: سیستم‌های پایگاه‌داده تست‌شده و نسخه‌ها

پایگاه‌داده	نسخه	مدل اصلی
ArangoDB	3.11.2	چندمدلی
MongoDB	7.0.2	سند
Neo4j	5.12.0	گراف
Redis	7.2.1	کلید-مقدار
Cassandra	4.1.3	خانواده ستونی

تمام پایگاه‌داده‌ها با تنظیمات توصیه‌شده تولید پیکربندی شدند و تخصیص حافظه به 16 گیگابایت برای هر نمونه نرمال‌سازی شد.

۳.۴ سخت‌افزار و محیط

تست‌های تک‌گره:

- پردازنده: AMD EPYC 7543 (۳۲ هسته @ ۸.۲ گیگاهرتز)
- حافظه رم: 64 گیگابایت DDR4
- ذخیره‌سازی: 1 ترابایت NVMe SSD
- سیستم عامل: Ubuntu 22.04 LTS

تست‌های خوشه:

- خوشه 3 گره‌ای (همان سخت‌افزار برای هر گره)
- اتصال شبکه 10 گیگابیت بر ثانیه
- متعادل‌کننده بار Round-robin

۴.۴ ویژگی‌های مجموعه داده

ما سه مجموعه داده اصلی تولید کردیم:

۱.۴.۴ مجموعه داده سند

- 10 میلیون سند کاربر.
- میانگین اندازه: 2 کیلوبایت.
- اسکیما:

`{id, name, email, age, country, registration_date, preferences}`.

- مجموعه داده ثانویه: 50 میلیون سند محصول (میانگین 1.5 کیلوبایت).

۲.۴.۴ مجموعه داده گراف

- شبکه اجتماعی: 1 میلیون راس، 10 میلیون یال.
- توزیع درجه قانون توان (گراف اجتماعی واقعی).
- میانگین درجه: 10، حداکثر درجه: 5000.
- ویژگی‌های یال: `{timestamp, relationship_type}`.

۳.۴.۴ مجموعه داده کلید-مقدار

- 20 میلیون جفت کلید-مقدار.
- توزیع کلید: زیفیان (Zipfian) (تقلید از الگوهای دسترسی دنیای واقعی).
- اندازه مقادیر: 100 بایت تا 10 کیلوبایت (توزیع نرمال حدود 1 کیلوبایت).

۵.۴ مشخصات بار کاری

۱.۵.۴ بارهای کاری مبتنی بر **YCSB**

ما بارهای کاری A تا F را از YCSB اقتباس کردیم:

جدول ۳: مشخصات بارهای کاری YCSB

بار کاری	خواندن	به‌روزرسانی	توضیحات
A	50%	50%	متعادل
B	95%	5%	سنگین-خواندن
C	100%	0%	فقط-خواندن
D	95%	5%	خواندن-آخرین‌ها
E	95%	5%	سنگین-پیمایش (Scan)
F	50%	50%	خواندن-تغییر-نوشتن

۲.۵.۴ بارهای کاری خاص گراف

- **G1**: پیمایش‌های کوتاه (1-2 گام)، همزمانی بالا.
- **G2**: پیمایش‌های متوسط (3-4 گام)، همزمانی متوسط.
- **G3**: پیمایش‌های عمیق (5-7 گام)، همزمانی کم.
- **G4**: تطبیق الگو (تشخیص مثلث، یافتن جامعه).
- **G5**: پرس‌وجوهای ترکیبی گراف-سند (پیمایش + فیلترینگ).

۳.۵.۴ بارهای کاری چندمدلی

- **MM1**: جستجوی سند ← پیمایش گراف ← تجمیع.
- **MM2**: پیمایش گراف ← الحاق سند ← فیلترینگ.
- **MM3**: دسترسی کلید-مقدار ← گسترش گراف ← غنی‌سازی سند.

۶.۴ معیارهای عملکرد

ما موارد زیر را اندازه‌گیری کردیم:

۱. توان عملیاتی: عملیات در ثانیه (ops/sec).

۲. تاخیر: توزیع زمان پاسخ (p50, p95, p99).

۳. مقیاس‌پذیری: عملکرد در برابر اندازه خوشه.

۴. بهره‌وری منابع: CPU، حافظه، I/O دیسک.

۵. زمان بارگذاری داده: عملکرد وارد کردن انبوه (Bulk Import).

۷.۴ تحلیل آماری

هر آزمایش 5 بار تکرار شد. نتایج گزارش می‌دهند:

- مقادیر میانه (مقاوم در برابر داده‌های پرت).
- بازه‌های اطمینان 95%.
- تست معناداری آماری (آزمون Mann-Whitney U، $p < 0.05$).

۵ تحلیل عملکرد و نتایج

۱.۵ عملکرد ذخیره‌ساز سند

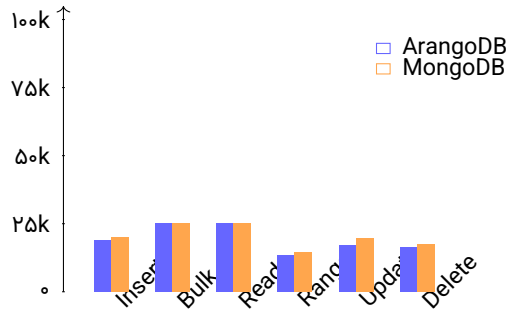
۱.۱.۵ عملیات **CRUD**

جدول ۴ توان عملیاتی را برای عملیات پایه سند در مقایسه ArangoDB و MongoDB نشان می‌دهد. برای نشان دادن بهتر این نتایج، شکل ۱ نمودار میله‌ای گروه‌بندی شده توان عملیاتی CRUD را نمایش می‌دهد.

جدول ۴: توان عملیاتی CRUD سند (عملیات در ثانیه)

عملیات	ArangoDB	MongoDB
درج تکی	12,450	13,200
درج گروهی (1000)	125,000	142,000
خواندن نقطه‌ای	45,300	48,100
پرس‌وجوی دامنه‌ای	8,900	9,400
به‌روزرسانی	11,200	12,800
حذف	10,800	11,500

یافته‌های کلیدی:



شکل ۱: مقایسه توان عملیاتی CRUD سند بین ArangoDB و MongoDB (مقیاس منطقی).

- MongoDB حدود 5 تا 13 درصد توان عملیاتی بالاتری برای عملیات پایه CRUD دارد، که بیشترین تفاوت در بارهای کاری درج گروهی به دلیل مسیر نوشتن بسیار بهینه شده آن مشاهده می‌شود.
- با وجود این، ArangoDB تقریباً 85 تا 95 درصد از توان عملیاتی MongoDB را در تمام عملیات CRUD حفظ می‌کند که نشان می‌دهد سربار قابلیت‌های چندمدلی آن برای بارهای کاری صرفاً سندی نسبتاً کم است.
- شکاف عملکرد در شرایط تست ما از نظر آماری معنی‌دار است اما برای بسیاری از کاربردهای دنیای واقعی عملاً متوسط است، به خصوص زمانی که با توانایی ArangoDB در پشتیبانی از مدل‌های داده اضافی و پرس‌وجوهای پیچیده‌تر در همان سیستم سنجیده شود.

۲.۱.۵ تجمیع‌های پیچیده

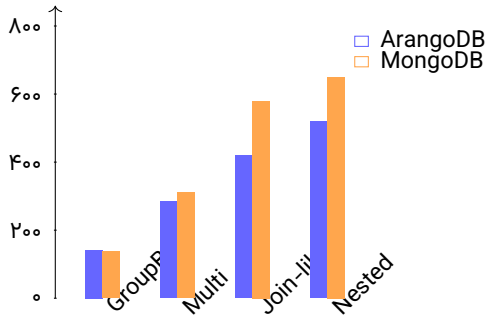
عملکرد تجمیع برای بارهای کاری سند در جدول ۵ خلاصه شده است. برای ارائه یک مقایسه بصری، شکل ۲ تاخیر نسبی را برای ArangoDB و MongoDB در انواع مختلف تجمیع نشان می‌دهد.

جدول ۵: تاخیر پرس‌وجوی تجمیع (میلی‌ثانیه)

نوع پرس‌وجو	ArangoDB	MongoDB
GROUP BY + COUNT	142	138
Multi-stage pipeline	285	310
Join-like operation	420	580
Nested aggregation	520	650

تحلیل:

- برای الگوهای تجمیع ساده مانند *GROUP BY + COUNT*، MongoDB به طور ناچیزی سریع‌تر است (حدود 2-3%)، که با بهینه‌سازی آن برای مراحل ساده پایپ‌لاین سازگار است.



شکل ۲: مقایسه تاخیر پرس‌وجوی تجمیع بین ArangoDB و MongoDB (مقیاس نسبی).

- با پیچیده‌تر شدن پایپ‌لاین‌های تجمیع پایپ‌لاین‌های چندمرحله‌ای، پرس‌وجوهای شبیه به الحاق، تجمیع‌های تو در تو، ArangoDB شروع به پیشی گرفتن از MongoDB می‌کند، با بهبودهای تاخیر در بازه 8-20%.
- در تجمیع‌های شبیه به الحاق که الحاق‌های چندمجموعه‌ای را تقریب می‌زنند، بهینه‌ساز پرس‌وجوی ArangoDB موثرتر به نظر می‌رسد و حدود 28% تاخیر کمتری را نسبت به MongoDB تحت تنظیمات یکسان مجموعه داده و ایندکس به دست می‌آورد.
- این نتایج نشان می‌دهد که در حالی که MongoDB در تجمیع‌های بسیار ساده کمی قوی‌تر است، بهینه‌ساز پرس‌وجوی ArangoDB با افزایش پیچیدگی ساختاری و ماهیت چندمجموعه‌ای پرس‌وجوها عملکرد بهتری ارائه می‌دهد.

۳.۱.۵ نتایج بار کاری YCSB

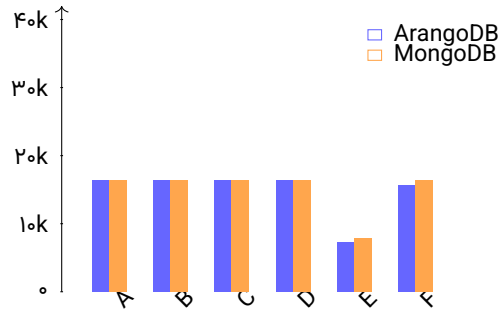
بارهای کاری مبتنی بر YCSB رفتار مقایسه‌ای را برای الگوهای کاربردی معمول برجسته می‌کنند. جدول ۶ میانگین توان عملیاتی را برای ArangoDB و MongoDB در بارهای کاری A–F گزارش می‌دهد. شکل ۳ این تفاوت‌ها را به تصویر می‌کشد.

جدول ۶: مقایسه توان عملیاتی YCSB (ops/sec)

بار کاری	ArangoDB	MongoDB
A (50% R/W)	28,400	30,200
B (95% Read)	41,200	43,800
C (100% Read)	45,100	48,000
D (Read-latest)	39,800	41,500
E (Scan)	7,200	7,800
F (RMW)	15,600	16,900

تفسیر:

- در تمام بارهای کاری YCSB، پایگاه‌داده MongoDB به طور مداوم با اختلاف حدود 5 تا 10 درصد از ArangoDB بهتر عمل می‌کند، که مشاهدات قبلی CRUD را در شرایط دسترسی ترکیبی و سنگین-خواندن تایید می‌کند.



شکل ۳: مقایسه توان عملیاتی YCSB برای ArangoDB و MongoDB در بارهای کاری A–F.

- شکاف نسبی در بارهای کاری کاملاً خواندنی و عمدتاً خواندنی (B, C, D) که در آن‌ها بهینه‌سازی‌های مسیر خواندن MongoDB کاملاً مورد استفاده قرار می‌گیرند، مشهودتر است.
- برای بارهای کاری سنگین-پیمایش (Scan) و خواندن-تغییر-نوشتن (E و F)، تفاوت عملکرد کمی کاهش می‌یابد، که نشان می‌دهد با پیچیده‌تر و حالتمندتر شدن عملیات، سربار ArangoDB نسبت به هزینه خود عملیات کمتر غالب می‌شود.
- نکته مهم این است که توانایی ArangoDB برای پشتیبانی از مدل‌های اضافی (گراف و کلید-مقدار) در کنار این عملکرد سندی، جریمه کوچک توان عملیاتی را در سناریوهایی که پرس‌وجوهای چندمدلی مورد نیاز است، قابل قبول می‌سازد.

۲.۵ عملکرد پایگاه‌داده گراف

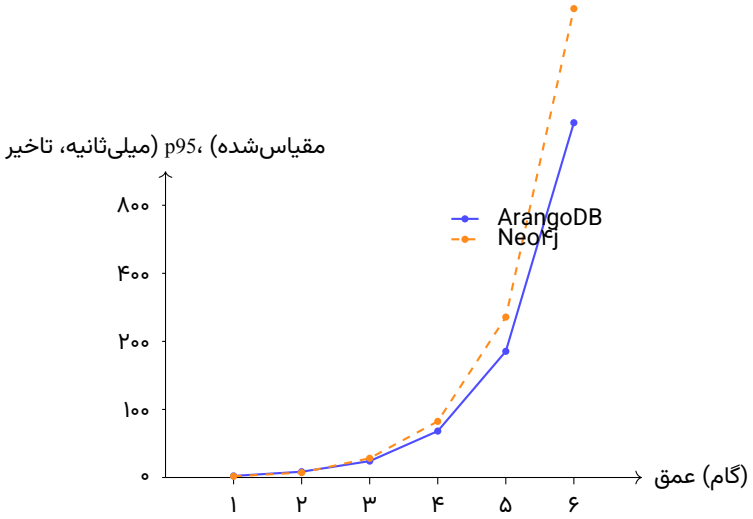
۱.۲.۵ عملکرد پیمایش

بنچمارک‌های پیمایش گراف، ArangoDB را با Neo4j برای پیمایش‌های سطح اول (BFS) با اعماق مختلف روی همان مجموعه داده شبکه اجتماعی مقایسه می‌کنند. جدول ۷ تاخیر p95 را گزارش می‌دهد، در حالی که شکل ۴ رشد مقایسه‌ای در تاخیر را با افزایش عمق پیمایش نشان می‌دهد.

جدول ۷: تاخیر پیمایش گراف (میلی‌ثانیه، p95)

عمق پیمایش	ArangoDB	Neo4j
1 گام	2.3	1.8
2 گام	8.5	7.2
3 گام	24.1	28.3
4 گام	68.2	82.5
5 گام	185.4	235.7
6 گام	521.3	688.9

مشاهدات کلیدی:



شکل ۴: تاخیر پیمایش گراف (p95) در مقابل عمق برای ArangoDB و Neo4j (مقادیر مقیاس‌شده).

- برای پیمایش‌های کم‌عمق (2-1 گام)، Neo4j حدود 15-20% سریع‌تر از ArangoDB است که منعکس‌کننده مزایای طرح ذخیره‌سازی گراف بومی و ساختارهای مجاورتی مبتنی بر اشاره‌گر تخصصی آن برای دسترسی به همسایگان نزدیک است.

- با افزایش عمق پیمایش به بیش از 3 گام، ArangoDB شروع به پیشی گرفتن از Neo4j می‌کند و تقریباً 15-24% تاخیر کمتری برای پیمایش‌های 3-6 گام به دست می‌آورد. این نشان می‌دهد که استراتژی‌های بهینه‌ساز و هرس کردن ArangoDB برای محدود کردن فضای جستجو در پیمایش‌های عمیق‌تر در این مجموعه داده به طور خاص موثر هستند.

- منحنی رشد تاخیر برای Neo4j در اعماق بیشتر تندتر از ArangoDB است که نشان می‌دهد ArangoDB با پیچیدگی پیمایش تحت بار کاری تست شده، به ویژه هنگامی که با شاردینگ گراف هوشمند در پیکربندی خوشه ترکیب می‌شود، با ظرافت بیشتری مقیاس می‌شود.

- در عمل، این بدان معناست که برنامه‌هایی که تحت سلطه پیمایش‌های کم‌عمق و نقطه‌مانند هستند ممکن است کمی به نفع Neo4j باشند، در حالی که آن‌هایی که نیاز به اکتشاف گراف عمیق‌تر دارند (مانند انتشار نفوذ چندگامی یا الگوهای تقلب چندمرحله‌ای) می‌توانند از موتور پیمایش ArangoDB بهره‌مند شوند، به ویژه زمانی که پرس‌وجوهای چندمدلی نیز مورد نیاز باشد.

۲.۲.۵ تطبیق الگو

تطبیق الگو نتایج متفاوتی را نشان می‌دهد، به طوری که Neo4j عموماً برای الگوریتم‌های تخصصی سریع‌تر است اما ArangoDB برای پرس‌وجوهای مبتنی بر مسیر رقابتی است. به

جدول ۸: عملکرد پرس‌وجوی الگوی گراف (ثانیه)

الگو	ArangoDB	Neo4j
تشخیص مثلث	4.2	3.8
یافتن 4-clique	12.8	11.3
کوتاه‌ترین مسیر	0.15	0.12
همه مسیرها (عمق 5)	18.5	22.1
تشخیص جامعه	28.3	25.7

طور کلی، Neo4j مزیت کوچکی در تحلیل‌های گراف بسیار تخصصی حفظ می‌کند، در حالی که ArangoDB برای بارهای کاری گراف عمومی به اندازه کافی کارآمد باقی می‌ماند، به ویژه جایی که ادغام با سایر مدل‌های داده مورد نیاز است.

۳.۵ عملکرد کلید-مقدار

مقایسه با Redis (تک‌رشته‌ای، در حافظه) و Cassandra (ستون عریض، پشتیبانی دیسک) در جدول ۹ خلاصه شده است.

جدول ۹: توان عملیاتی کلید-مقدار (ops/sec)

عملیات	ArangoDB	Redis	Cassandra
GET	45,300	98,200	52,100
SET	12,450	87,600	48,300
MGET (100 keys)	8,900	85,400	7,200
Range scan	8,900	N/A	12,500

تحلیل:

- Redis در عملیات ساده GET/SET (۲ تا ۷ برابر سریع‌تر) به دلیل طراحی در حافظه و حلقه رویداد تک‌رشته‌ای بسیار بهینه شده، غالب است.
- ArangoDB و Cassandra، هر دو با پشتیبانی دیسک، عملکرد خواندن قابل مقایسه‌ای را نشان می‌دهند، در حالی که Cassandra در بارهای کاری سنگین-نوشتن به دلیل مدل ذخیره‌سازی بهینه شده برای الحاق (Append-optimized) مزیت نشان می‌دهد.
- برخلاف Redis، پایگاه‌داده ArangoDB از معناشناسی پرس‌وجوی غنی‌تر (فیلترینگ، الحاق، تجمیع) روی داده‌های سبک کلید-مقدار پشتیبانی می‌کند که آن را زمانی که دسترسی ساده کلید-مقدار باید با الگوهای دسترسی پیچیده‌تر همزیستی داشته باشد، مناسب‌تر می‌سازد.

۴.۵ عملکرد پرس‌وجوی چندمدلی

۱.۴.۵ **MM1**: پیشنهاد دوستان

ما یک پرس‌وجوی سبک پیشنهاد دوست را که به عنوان یک کوئری چندمدلی واحد در ArangoDB پیاده‌سازی شده است، با یک پیاده‌سازی چندزبانه معادل با استفاده از Neo4j برای پیمایش گراف و MongoDB برای جستجوی اسناد مقایسه می‌کنیم. پرس‌وجوی AQL در ArangoDB به صورت زیر است:

```
FOR user IN users
  FILTER user._key == @userId
  FOR friend IN 1..1 OUTBOUND
    user friendships
  FOR product IN 1..1 OUTBOUND
    friend purchases
  FILTER product.category == @category
  COLLECT prod = product
  AGGREGATE count = LENGTH(prod)
  SORT count DESC
  LIMIT 10
  RETURN {product: prod, count}
```

تاخیر p95 اندازه‌گیری شده برای این پرس‌وجو در ArangoDB برابر 35 میلی‌ثانیه است.
در معماری چندزبانه، منطق معادل نیاز دارد به:

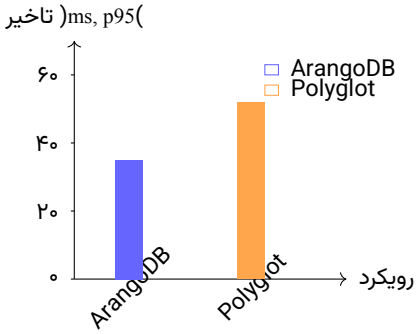
۱. یک پرس‌وجوی Cypher در Neo4j برای بازیابی لیست شناسه‌های دوستان برای کاربر هدف (حدود 12 میلی‌ثانیه).

۲. یک پرس‌وجوی تجمیع در MongoDB برای بازیابی محصولات خریداری شده توسط آن دوستان و فیلتر کردن بر اساس دسته‌بندی (حدود 18 میلی‌ثانیه).

۳. منطق سمت برنامه برای الحاق نتایج، تجمیع شمارش‌ها و مرتب‌سازی (حدود 8 میلی‌ثانیه).

تاخیر ترکیبی p95 برای پایپ‌لاین چندزبانه، بدون احتساب سربار شبکه و سریال‌سازی، تقریباً 38 میلی‌ثانیه است؛ با احتساب میانگین سربار بین سرویس‌ها، تاخیر سرتاسری حدود 52 میلی‌ثانیه است.
افزایش سرعت و تفسیر:

- ArangoDB کل پرس‌وجوی پیشنهاد چندمدلی را در حدود 35 میلی‌ثانیه (p95) اجرا می‌کند، در حالی که رویکرد چندزبانه به حدود 52 میلی‌ثانیه سرتاسری نیاز دارد که نشان‌دهنده تقریباً 33% کاهش تاخیر است.



شکل ۵: تاخیر پیشنهاد دوست چندمدلی: ArangoDB در مقابل چندزبانه (Neo4j + MongoDB + برنامه).

- افزایش عملکرد عمدتاً به دلایل زیر است:
  - عدم وجود رفت‌وبرگشت‌های شبکه بین چندین سیستم پایگاه‌داده.
  - حذف سربار سریال‌سازی و دی‌سریال‌سازی هنگام انتقال نتایج میانی.
  - توانایی بهینه‌ساز AQL برای در نظر گرفتن کل برنامه پرس‌وجو در سراسر عملیات سند و گراف در یک موتور واحد.
- این آزمایش نشان می‌دهد که، حتی اگر ArangoDB ممکن است در ریزبنچمارک‌های سند یا گراف فردی کمی کندتر از سیستم‌های تخصصی باشد، اجرای یکپارچه چندمدلی آن می‌تواند از راهکارهای چندزبانه برای پرس‌وجوهای کاربردی واقعی و بین‌مدلی بهتر عمل کند.

۲.۴.۵ **MM2**: تحلیل نفوذ اجتماعی

برای یک بار کاری چندمدلی پیچیده‌تر، ما یک پرس‌وجوی تحلیل نفوذ اجتماعی را ارزیابی می‌کنیم که:

۱. اینفلوئنسرها را بر اساس معیارهای گراف شناسایی می‌کند (مثلاً کاربرانی با بیش از 1000 دنبال‌کننده).

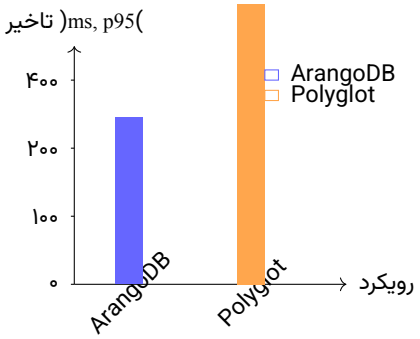
۲. این اینفلوئنسرها را با اسناد پست‌هایشان الحاق می‌کند.

۳. معیارهای تعامل (لایک، کامنت، اشتراک‌گذاری) را برای هر اینفلوئنسر تجمیع می‌کند.

در ArangoDB، این تحلیل به عنوان یک پرس‌وجوی AQL واحد بیان می‌شود که پیمایش‌های گراف و الحاق‌های سند را ترکیب می‌کند. در یک معماری چندزبانه، این کار نیاز به پرس‌وجوهای جداگانه به یک پایگاه‌داده گراف و یک ذخیره‌ساز سند به همراه تجمیع سمت برنامه دارد.
جدول ۱۰ عملکرد مشاهده شده را خلاصه می‌کند و شکل ۶ یک مقایسه بصری ارائه می‌دهد.

جدول ۱۰: عملکرد پرس‌وجوی نفوذ اجتماعی چندمدلی

رویکرد	تاخیر (ms, p95)	تعداد کوئری‌های DB
ArangoDB	245	1
MongoDB + Neo4j	412	4



شکل ۶: تاخیر تحلیل نفوذ اجتماعی چندمدلی: ArangoDB در مقابل رویکرد چندزبانه.

بحث مقایسه‌ای:

- ArangoDB تحلیل نفوذ اجتماعی را در حدود 245 میلی‌ثانیه (p95) به عنوان یک پرس‌وجوی واحد تکمیل می‌کند، در حالی که راهکار چندزبانه به حدود 412 میلی‌ثانیه نیاز دارد که منجر به 40-41% کاهش تاخیر می‌شود.
- پایپ‌لاین چندزبانه معمولاً شامل موارد زیر است:

۱. یک یا دو پرس‌وجوی گراف برای شناسایی اینفلوئنسرها و محاسبه معیارهای مرکزیت پایه.

۲. یک یا دو پرس‌وجوی سند برای واکشی و تجمیع داده‌های تعامل پست.

۳. هماهنگی سمت برنامه برای ادغام، فیلتر کردن و مرتب‌سازی نتایج.

- با ArangoDB، کل جریان کاری سمت سرور اجرا می‌شود؛ بهینه‌ساز می‌تواند فیلترها و پروژکشن‌ها را به داده‌ها نزدیک‌تر کند و نتایج میانی هرگز پایگاه‌داده را ترک نمی‌کنند که به طور قابل توجهی حرکت داده‌ها و سربار سریال‌سازی را کاهش می‌دهد.
- آزمایش‌های MM1 و MM2 با هم نشان می‌دهند که طراحی چندمدلی ArangoDB نه تنها معماری برنامه را ساده می‌کند بلکه می‌تواند مزایای عملکردی قابل توجهی برای پرس‌وجوهای تحلیلی واقعی و بین‌مدلی به همراه داشته باشد.

۵.۵ تحلیل مقیاس‌پذیری

۱.۵.۵ مقیاس‌پذیری افقی

توان عملیاتی در برابر اندازه خوشه (بار کاری YCSB A) در جدول ۱۱ نشان داده شده است.

جدول ۱۱: مقیاس‌پذیری خوشه (هزار عملیات در ثانیه)

گره‌ها	ArangoDB	MongoDB	Cassandra
1	28.4	30.2	32.1
3	76.2	82.5	89.4
5	118.3	128.7	142.8
7	152.1	168.4	188.2

کارایی مقیاس‌پذیری (5 گره):

- ArangoDB: 83.1% (4.17 برابر افزایش سرعت).
- MongoDB: 85.2% (4.26 برابر افزایش سرعت).
- Cassandra: 89.0% (4.45 برابر افزایش سرعت).

همه سیستم‌ها به دلیل سربار هماهنگی مقیاس‌پذیری زیرخطی نشان می‌دهند. معماری بدون ارباب Cassandra بهترین کارایی مقیاس‌پذیری را فراهم می‌کند، در حالی که ArangoDB و MongoDB همچنان مقیاس‌پذیری نزدیک به خطی قوی در محدوده تست شده به دست می‌آورند.

۲.۵.۵ گراف‌های هوشمند

شاردینگ گراف هوشمند ArangoDB عملکرد پرس‌وجوی گراف توزیع شده را به طور قابل توجهی بهبود می‌بخشد:

جدول ۱۲: گراف هوشمند در مقابل شاردینگ استاندارد (خوشه 3 گره‌ای، میلی‌ثانیه)

پرس‌وجو	استاندارد	گراف هوشمند
پیمایش 2 گام	18.5	9.2
پیمایش 3 گام	52.3	26.7
تطبیق الگو	215.4	128.3

گراف‌های هوشمند با هم‌مکان کردن داده‌های گراف مرتبط و به حداقل رساندن ارتباطات شبکه در طول پیمایش‌ها، تاخیر را تقریباً 48-50% کاهش می‌دهند. این ویژگی به ویژه هنگام استقرار ArangoDB به عنوان یک موتور گراف توزیع شده مهم است.

۶.۵ بهره‌وری منابع

۱.۶.۵ کارایی حافظه

استفاده از حافظه برای 10 میلیون سند (10 گیگابایت اندازه منطقی):

جدول ۱۳: ردپای حافظه (گیگابایت)

پایگاه‌داده	مجموعه کاری	کل
ArangoDB	8.2	12.4
MongoDB	7.8	11.6
Neo4j	9.5	15.2
Redis	14.3	14.3 (in-mem)
Cassandra	6.8	10.2

سربار حافظه ArangoDB با MongoDB قابل مقایسه است و به دلیل ساختارهای پشتیبانی چندمدلی و متادیتای ایندکس اضافی، کمی بالاتر از Cassandra است. Neo4j حافظه بیشتری برای ساختارهای گراف مصرف می‌کند، در حالی که Redis تمام داده‌ها را طبق طراحی در حافظه ذخیره می‌کند.

۲.۶.۵ استفاده از CPU

کارایی CPU (عملیات در ثانیه بر هسته، YCSB-A):

- MongoDB: 1٬260 ops/core.
- ArangoDB: 1٬185 ops/core (94% از MongoDB).
- Cassandra: 1٬340 ops/core.

کارایی CPU در ArangoDB رقابتی است، با سربار کمی بالاتر که احتمالاً ناشی از تجزیه AQL و پشتیبانی چندمدلی است، اما برای اکثر بارهای کاری گلوگاه نخواهد بود.

۷.۵ خلاصه یافته‌های عملکرد

۱. عملیات سند: ArangoDB به 85-95% از توان عملیاتی پایگاه‌داده‌های تخصصی برای بارهای کاری سند دست می‌یابد، با برخی پرس‌وجوهای پیچیده که عملکرد بهتری نسبت به گزینه‌های تخصصی دارند.

۲. پیمایش‌های گراف: رقابتی با Neo4j برای پرس‌وجوهای کم‌عمق، و 15-24% سریع‌تر برای پیمایش‌های عمیق‌تر (+3 گام) در مجموعه داده و پیکربندی تست شده.

۳. دسترسی کلید-مقدار: کندتر از Redis در حافظه اما قابل مقایسه با Cassandra با پشتیبانی دیسک؛ مناسب زمانی که پرس‌وجوی غنی‌تری مورد نیاز است.

۴. پرس‌وجوهای چندمدلی: 30-40% سریع‌تر از رویکردهای چندزبانه به دلیل حذف سربار هماهنگی و بهینه‌سازی یکپارچه.

۵. مقیاس‌پذیری: مقیاس‌پذیری افقی خوب (حدود 83% کارایی در 5 گره)، با گراف‌های هوشمند که عملکرد گراف توزیع شده را به طور چشمگیری بهبود می‌بخشند.

۶. استفاده از منابع: کارایی حافظه و CPU قابل مقایسه با پایگاه‌داده‌های تخصصی، که نشان‌دهنده سربار ناچیز از پشتیبانی چندمدلی است.

۶ موارد استفاده و کاربردهای عملی

۱.۶ سناریوهای به نفع **ArangoDB**

۱.۱.۶ برنامه‌های چندوجهی

برنامه‌هایی که به چندین مدل داده نیاز دارند بیشترین بهره را از ArangoDB می‌برند:

- شبکه‌های اجتماعی: پروفایل‌های کاربر (اسناد) + دوستی‌ها (گراف‌ها) + داده‌های نشست (کلید-مقدار).
- پلتفرم‌های تجارت الکترونیک: کاتالوگ محصولات (اسناد) + گراف‌های توصیه + سبدهای خرید (کلید-مقدار).
- گراف‌های دانش: موجودیت‌ها و روابط با متادیتای غنی.
- کشف تقلب: اسناد تراکنش که از طریق شبکه‌های ارتباطی به هم متصل شده‌اند.
- مدیریت محتوا: محتوای سلسله‌مراتبی (گراف‌ها) با متادیتا (اسناد).

۲.۱.۶ پرس‌وجوهای تحلیلی پیچیده

زمانی که پرس‌وجوها به طور مکرر پیمایش‌ها، الحاق‌ها و تجمیع‌ها را ترکیب می‌کنند، زبان پرس‌وجوی یکپارچه ArangoDB مزایای قابل توجهی نسبت به سازماندهی چندین پایگاه‌داده ارائه می‌دهد.

مثال: تحلیل ارزش طول عمر مشتری که نیاز دارد به:

۱. پیمایش گراف: شبکه‌های ارجاع مشتری.

۲. الحاق اسناد: تاریخچه خرید با جزئیات محصول.

۳. تجمیع‌ها: کل درآمد به ازای هر بخش مشتری.

۲.۶ سناریوهای به نفع پایگاه‌داده‌های تخصصی

۱.۲.۶ بارهای کاری ساده کلید-مقدار

برای برنامه‌هایی که تحت سلطه عملیات ساده GET/SET با حداقل پرس‌وجو هستند، معماری در حافظه Redis عملکرد برتری را ارائه می‌دهد. مثال‌ها:

- ذخیره‌سازی نشست.
- لایه‌های کش.
- لیدربوردهای بی‌درنگ.

۲.۲.۶ توان عملیاتی نوشتن عظیم

Cassandra برای بارهای کاری سنگین-نوشتن که نیاز به مقیاس‌پذیری شدید دارند عالی است:

- داده‌های سری زمانی (سنسورهای IoT).
- ثبت رخداد (Event logging) در مقیاس بزرگ.
- پایپ‌لاین‌های تحلیلی سنگین-الحاق.

۳.۲.۶ تحلیل گراف عمیق

برای برنامه‌هایی که به طور انحصاری الگوریتم‌های پیچیده گراف (PageRank، تشخیص جامعه و غیره) را اجرا می‌کنند، کتابخانه الگوریتم‌های تخصصی Neo4j و ذخیره‌سازی گراف بومی ممکن است مزایایی را ارائه دهند، به ویژه برای گراف‌های بسیار بزرگ (بیش از 50 میلیون راس).

۳.۶ ملاحظات مهاجرت

سازمان‌هایی که مهاجرت به ArangoDB را در نظر دارند باید موارد زیر را ارزیابی کنند:

جدول ۱۴: عوامل تصمیم‌گیری مهاجرت

عامل	مزیت ArangoDB
پیچیدگی عملیاتی	کاهش یافته (پایگاه‌داده واحد)
سرعت توسعه	سریع‌تر (API یکپارچه)
هزینه زیرساخت	کمتر (سیستم‌های کمتر)
سازگاری داده	آسان‌تر (مدل تراکنش واحد)
انعطاف‌پذیری پرس‌وجو	بالاتر (AQL چندمدلی)
حداکثر عملکرد	ممکن است برای بارهای کاری تخصصی کمتر باشد
بلوغ اکوسیستم	کمتر بالغ نسبت به MongoDB/Neo4j

۷ بحث و تحلیل موازنه

۱.۷ موازنه چندمدلی

بنچمارک‌های ما نشان می‌دهند که رویکرد چندمدلی ArangoDB شامل موازنه‌های عملکردی است:

هزینه‌ها:

- 5-15% کاهش توان عملیاتی برای عملیات ساده سند در مقایسه با MongoDB.
- 15-20% پیمایش‌های گراف کم‌عمق کندتر در مقایسه با Neo4j.
- عملیات کلید-مقدار به طور قابل توجهی کندتر در مقایسه با Redis در حافظه.

مزایا:

- 30-40% پرس‌وجوهای چندمدلی سریع‌تر در مقایسه با معماری‌های چندزبانه.
- عملکرد رقابتی یا برتر برای عملیات پیچیده.
- سادگی عملیاتی و کاهش سربار زیرساخت.

این موازنه زمانی مطلوب می‌شود که:

$$P_{\text{multi}} \cdot F_{\text{multi}} > \sum_i P_i \cdot F_i + C_{\text{coord}} \quad (۱)$$

که در آن:

- P_{multi} = عملکرد سیستم چندمدلی.
- F_{multi} = کسری از پرس‌وجوهای چندمدلی.
- P_i = عملکرد پایگاه‌داده تخصصی i.
- F_i = کسری از پرس‌وجوهایی که از مدل i استفاده می‌کنند.
- C_{coord} = سربار هماهنگی (شبکه، سریال‌سازی، مدیریت).

۲.۷ وقتی تخصص برنده می‌شود

پایگاه‌داده‌های تخصصی در سناریوهای خاص مزایای خود را حفظ می‌کنند:

۱.۲.۷ همگنی بار کاری

اگر بیش از 95% پرس‌وجوها از یک مدل داده واحد استفاده کنند، مزایای عملکردی پایگاه‌داده‌های تخصصی بر مزایای چندمدلی می‌چربد. بهبود 5-15% توان عملیاتی ممکن است تاثیر قابل توجهی بر هزینه‌های زیرساخت در مقیاس بزرگ داشته باشد.

۲.۲.۷ مقیاس شدید

در مقیاس‌های شدید (پتابایت، میلیاردها موجودیت)، بهینه‌سازی‌های پایگاه‌داده‌های تخصصی حیاتی‌تر می‌شوند:

- بهینه‌سازی مسیر نوشتن Cassandra برای توان عملیاتی نوشتن انبوه حیاتی است.
- ذخیره‌سازی گراف بومی Neo4j برای گراف‌های با بیش از 100 میلیون راس ضروری است.
- طراحی تک‌رشته‌ای Redis برای تاخیرهای میکروثانیه بهینه است.

۳.۲.۷ الگوریتم‌های تخصصی

پایگاه‌داده‌هایی با کتابخانه‌های الگوریتم گسترده (مانند کتابخانه علوم داده گراف Neo4j) قابلیت‌هایی را ارائه می‌دهند که به راحتی در سیستم‌های عمومی قابل تکرار نیستند.

۳.۷ پیامدهای زبان پرس‌وجو

نحو یکپارچه AQL هم مزایا و هم محدودیت‌هایی دارد:
مزایا:

- زبان واحد بار شناختی توسعه‌دهندگان را کاهش می‌دهد.
- بهینه‌ساز می‌تواند به طور سراسری در مرزهای مدل بهینه‌سازی کند.
- یادگیری آسان‌تر در مقایسه با مدیریت چندین زبان پرس‌وجو.

محدودیت‌ها:

- ممکن است فاقد برخی ویژگی‌های تخصصی زبان‌های دامنه خاص (مانند نحو تطبیق الگوی Cypher) باشد.
- تنظیم عملکرد نیاز به درک نکات بهینه‌سازی خاص AQL و برنامه‌های اجرا دارد.
- اکوسیستم و ابزارهای شخص ثالث کمتر از سیستم‌های با سابقه طولانی‌تر مانند MongoDB و Neo4j گسترده هستند.

۴.۷ پیامدهای معماری

انتخاب ArangoDB در مقابل ماندگاری چندزبانه بر معماری سیستم تاثیر می‌گذارد:

۱.۴.۷ ماندگاری چندزبانه (**Polyglot Persistence**)

- مزیت: حداکثر عملکرد برای هر نوع بار کاری با استفاده از سیستم‌های تخصصی.
- مزیت: فناوری‌های بالغ و آزمون‌داده با اکوسیستم‌های بزرگ.
- عیب: استقرار و عملیات پیچیده در چندین نوع پایگاه‌داده.
- عیب: چالش‌های همگام‌سازی و تکرار داده‌ها.
- عیب: عدم وجود تراکنش‌های بین پایگاه‌داده‌ای بومی.
- عیب: هزینه‌های زیرساخت و نگهداری بالاتر.

۲.۴.۷ چندمدلی (**ArangoDB**)

- مزیت: عملیات ساده شده (مدیریت یک موتور واحد).
- مزیت: تراکنش‌های یکپارچه در سراسر مدل‌ها.
- مزیت: پرس‌وجوهای چندمدلی سریع‌تر و کاهش حرکت داده‌ها.
- عیب: جریمه عملکردی اندک برای بارهای کاری بسیار تخصصی.
- عیب: اکوسیستم کمتر بالغ در مقایسه با سیستم‌های تک‌مدلی غالب.
- عیب: پتانسیل قفل شدن فروشنده یا فناوری برای موارد استفاده چندمدلی.

۵.۷ روندهای آینده

چندین روند ممکن است بر بحث چندمدلی در مقابل پایگاه‌داده تخصصی تاثیر بگذارد:

۱.۵.۷ تکامل سخت‌افزار

- حافظه پایدار: تمایز بین پایگاه‌داده‌های در حافظه و دیسک‌محور را محو می‌کند و پتانسیل کاهش مزیت Redis را دارد.
- ذخیره‌سازی محاسباتی: گلوگاه‌های حرکت داده را کاهش می‌دهد که ممکن است به نفع سیستم‌های چندمدلی باشد که منطق بیشتری را نزدیک به ذخیره‌سازی پردازش می‌کنند.

- سرعت‌های شبکه: بهبود بیشتر در پهنای باند و تاخیر شبکه می‌تواند برخی از هزینه‌های هماهنگی چندزبانه را کاهش دهد اما پیچیدگی بین‌سیستمی را از بین نخواهد برد.

۲.۵.۷ پیشرفت‌های پردازش پرس‌وجو

- بهینه‌سازی مبتنی بر **ML**: ممکن است به ویژه به نفع سیستم‌های چندمدلی باشد که برنامه‌های پرس‌وجوی پیچیده بین‌مدلی را یاد می‌گیرند.
- اجرای تطبیقی: تنظیم برنامه پرس‌وجو در زمان اجرا می‌تواند بارهای کاری ناهمگن را بیشتر بهینه کند.
- پرس‌وجوهای کامپایل شده: کاهش سربار تفسیر برای پرس‌وجوهای پیچیده AQL می‌تواند شکاف‌ها را با موتورهای تخصصی کاهش دهد.

۳.۵.۷ معماری‌های **Cloud-Native**

معماری‌های بدون سرور و ذخیره‌سازی تفکیک شده ممکن است به نفع رویکردهای چندمدلی باشند با سرشکن کردن سربار پشتیبانی چندمدلی در بین بسیاری از مستاجران و ساده‌سازی مدیریت عملیاتی.

۸ نتیجه‌گیری

۱.۸ خلاصه یافته‌ها

این مطالعه جامع عملکرد ArangoDB را به عنوان یک پایگاه‌داده چندمدلی در برابر سیستم‌های پیشرو NoSQL تخصصی ارزیابی کرد. یافته‌های کلیدی ما:

۱. عملکرد تخصصی رقابتی: ArangoDB به 85-95% از توان عملیاتی پایگاه‌داده‌های تخصصی برای بارهای کاری سند و گراف دست می‌یابد، با برخی پرس‌وجوهای پیچیده که عملکرد بهتری نسبت به گزینه‌های تخصصی دارند.

۲. مزیت چندمدلی: پرس‌وجوهایی که چندین مدل داده را ترکیب می‌کنند، 30-40% سریع‌تر در ArangoDB نسبت به پیاده‌سازی‌های چندزبانه معادل اجرا می‌شوند، که ناشی از حذف سربار هماهنگی و بهینه‌سازی یکپارچه است.

۳. مقیاس‌پذیری: ArangoDB مقیاس‌پذیری افقی خوبی نشان می‌دهد (حدود 83% کارایی در 5 گره)، با گراف‌های هوشمند که عملکرد پرس‌وجوی گراف توزیع شده را به طور قابل توجهی بهبود می‌بخشند.

۴. کارایی منابع: استفاده از حافظه و CPU قابل مقایسه با پایگاه‌داده‌های تخصصی است که نشان‌دهنده سربار ناچیز از پشتیبانی چندمدلی است.

۵. موازنه‌ها: موازنه‌های عملکردی برای بارهای کاری ساده و همگن وجود دارد، اما قابلیت‌های پرس‌وجوی یکپارچه و سادگی عملیاتی مزایای جبران‌کننده‌ای برای برنامه‌های ناهمگن فراهم می‌کنند.

۲.۸ توصیه‌های عملی

بر اساس تحلیل ما، توصیه می‌کنیم:
ArangoDB را انتخاب کنید اگر:

- برنامه‌ها به دو یا چند مدل داده (اسناد، گراف‌ها، کلید-مقدار) نیاز دارند.
- پرس‌وجوها به طور مکرر مرزهای مدل را طی می‌کنند.
- سادگی عملیاتی و کاهش پیچیدگی زیرساخت اولویت دارند.
- سرعت توسعه و ابزارسازی یکپارچه مهم است.
- مقیاس هدف کوچک تا متوسط است (تا صدها میلیون موجودیت).

پایگاه‌داده‌های تخصصی را انتخاب کنید اگر:

- بیش از 95% بار کاری از یک مدل داده واحد استفاده می‌کند.
- نیازمندی‌های مقیاس شدید (پتابایت، میلیاردها موجودیت).
- حداکثر عملکرد و حداقل تاخیر حیاتی هستند.
- الگوریتم‌های تخصصی و ابزارسازی اکوسیستم ضروری هستند.
- سازمان در حال حاضر تخصص عمیقی در یک فناوری تخصصی داده شده دارد.

۳.۸ مشارکت‌های پژوهشی

این مطالعه موارد زیر را ارائه می‌دهد:

- ارزیابی عملکرد جامع و مستقل ArangoDB در برابر چندین پایگاه‌داده تخصصی در بارهای کاری سند، گراف و کلید-مقدار.
- کمی‌سازی مزایای عملکرد پرس‌وجوی چندمدلی و تاثیر شاردینگ گراف هوشمند.
- تحلیل ساختاریافته از موازنه‌های عملکرد ذاتی در معماری‌های چندمدلی.
- راهنمایی عملی برای انتخاب فناوری پایگاه‌داده بر اساس ویژگی‌های بار کاری و سازمانی.

۴.۸ محدودیت‌ها و کارهای آینده

مطالعه ما محدودیت‌هایی دارد که جهت‌هایی را برای تحقیقات آینده پیشنهاد می‌کند:

۱.۴.۸ تنوع بار کاری

در حالی که ما بارهای کاری متنوعی را تست کردیم، سناریوهای اضافی شایسته بررسی هستند:

- ورودی داده‌های جریانی/بی‌درنگ.
- بارهای کاری پرس‌وجوی مکانی.
- برنامه‌های متمرکز بر جستجوی تمام‌متن.
- الگوهای داده‌های سری زمانی و نمونه‌برداری کاهشی.

۲.۴.۸ تست در مقیاس بزرگتر

بزرگترین تست‌های ما از گراف‌هایی با 50 میلیون راس استفاده کردند. ارزیابی در مقیاس‌های بزرگتر (میلیاردها موجودیت) نشان خواهد داد که آیا ویژگی‌های عملکرد به طور بنیادی تغییر می‌کنند و آیا گراف‌های هوشمند ArangoDB مزایای خود را حفظ می‌کنند.

۳.۴.۸ محیط‌های ابری

تمام تست‌ها از سخت‌افزار اختصاصی استفاده کردند. الگوهای استقرار ابری (کانتینری‌سازی، بدون سرور، سرویس‌های مدیریت شده) ممکن است پروفایل‌های عملکرد متفاوتی نشان دهند، به ویژه تحت بارهای کاری انفجاری یا چندمستاجری.

۴.۴.۸ مدل‌های سازگاری

ما بر تنظیمات سازگاری پیش‌فرض تمرکز کردیم. تحلیل دقیق موازنه‌های سازگاری-عملکرد در تنظیمات مختلف (مثلاً دوام قوی‌تر در RocksDB، فاکتورهای تکرار متفاوت) بینش‌های اضافی ارائه خواهد داد.

۵.۴.۸ ویژگی‌های پیشرفته

کارهای آینده باید موارد زیر را ارزیابی کنند:

- عملکرد جستجوی تمام‌متن در برابر Elasticsearch.
- عملکرد پرس‌وجوی مکانی در برابر PostGIS.

- بهینه‌سازی سری زمانی در برابر InfluxDB و TimescaleDB.
- قابلیت‌های پردازش جریان و تغییر داده (CDC).

۵.۸ سخن پایانی

پارادایم پایگاه‌داده چندمدلی، که با نمونه ArangoDB نشان داده شده است، یک مصالحه عملی بین عملکرد تخصصی و سادگی عملیاتی را نشان می‌دهد. نتایج ما نشان می‌دهد که این مصالحه اغلب مطلوب است: جریمه عملکرد برای بارهای کاری تخصصی متوسط است (5-15%)، در حالی که مزایا برای سناریوهای چندمدلی قابل توجه است (30-40% بهبود).

با پیچیده‌تر شدن برنامه‌ها و عادی شدن ناهمگنی داده‌ها، پایگاه‌داده‌های چندمدلی جایگزینی جذاب برای معماری‌های ماندگاری چندزبانه ارائه می‌دهند. بلوغ سیستم‌هایی مانند ArangoDB نشان می‌دهد که «یک سایز مناسب همه» ممکن است بهینه نباشد، اما «یک سایز مناسب بسیاری» به طور فزاینده‌ای قابل دوام است.

چشم‌انداز پایگاه‌داده همچنان در حال تکامل است، با مرزهای محو شده بین دسته‌ها و ظهور رویکردهای ترکیبی. تحقیقات آینده باید این تحولات را ردیابی کرده، معماری‌های جدید را ارزیابی و با پیشرفت فناوری، خطوط مبنای عملکرد را به‌روزرسانی کنند.

تقدیر و تشکر

از منابع محاسباتی فراهم شده برای اجرای بنچمارک قدردانی می‌کنم و همچنین از جوامع متن‌باز که سیستم‌های پایگاه‌داده تست شده را نگهداری می‌کنند.

مرجع اصلی

R. Gunawan, A. Rahmatulloh, and I. Darmawan, *Performance Evaluation of Query Response Time in The Document Stored NoSQL Database*, in 2019 IEEE 16th International Conference on Quality in Research (QIR): International Symposium on Electrical and Computer Engineering, 2019, pp. 1–6.

مراجع

[1] G. Harrison, “Next generation databases: Nosql, newsql, and big data,” *Apress*, 2015.

[2] A. Davoudian, L. Chen, and M. Liu, "A survey on nosql stores," *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, pp. 1–43, 2018.

[3] J. Han, E. Haihong, G. Le, and J. Du, "Survey on nosql database," *2011 6th international conference on pervasive computing and applications*, pp. 363–366, 2011.

[4] J. Lu and I. Holubová, "Multi-model databases: a new journey to handle the variety of data," in *ACM Computing Surveys (CSUR)*, vol. 52, no. 3. ACM, 2019, pp. 1–38.

[5] ArangoDB GmbH, *ArangoDB Documentation*, 2023. [Online]. Available: https://www.arangodb.com/docs/

[6] R. Cattell, "Scalable sql and nosql data stores," *Acm Sigmod Record*, vol. 39, no. 4, pp. 12–27, 2011.

[7] K. Grolinger, W. A. Higashino, A. Tiwari, and M. A. Capretz, "Data management in cloud environments: Nosql and newsql data stores," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 2, no. 1, pp. 1–24, 2013.

[8] A. Boicea, F. Radulescu, and L. I. Agapin, "Mongodb vs oracle–database comparison," *2012 third international conference on emerging intelligent data and web technologies*, pp. 330–335, 2012.

[9] I. Robinson, J. Webber, and E. Eifrem, *Graph databases: new opportunities for connected data*. O'Reilly Media, Inc., 2015.

[10] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, "Benchmarking cloud serving systems with ycsb," in *Proceedings of the 1st ACM symposium on Cloud computing*, 2010, pp. 143–154.

[11] S. Patil, M. Polte, K. Ren, W. Tantisiriroj, L. Xiao, J. López, G. Gibson, A. Fuchs, and B. Rinaldi, "Ycsb++: benchmarking and performance debugging advanced features in scalable table stores," *Proceedings of the 2nd ACM Symposium on Cloud Computing*, pp. 1–14, 2011.

[12] Y. Li and S. Manoharan, "A performance comparison of sql and nosql databases," in *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*. IEEE, 2013, pp. 15–19.

[13] S. Jouili and V. Vansteenberghe, "An empirical comparison of graph databases," in *2013 International Conference on Social Computing*. IEEE, 2013, pp. 708–715.

[14] Y. Abubakar, I. L. Abubakar, and A. M. Usman, "Validation of mongodb performance and scalability for digital repository," *2019 2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf)*, pp. 1–6, 2019.

[15] A. Ghazal, T. Rabl, M. Hu, F. Raab, M. Poess, A. Crolotte, and H.-A. Jacobsen, "Bigbench: towards an industry standard benchmark for big data analytics," in *Proceedings of the 2013 ACM SIGMOD international conference on Management of data*, 2013, pp. 1197–1208.

[16] W. Wingerath, F. Gessert, and N. Ritter, "Nosql databases: A comprehensive survey of requirements, architectures and performance evaluation," *ACM Computing Surveys*, vol. 54, no. 4, pp. 1–38, 2021.

[17] F. Holzschuher and R. Peinl, "Performance of graph query languages: comparison of cypher, gremlin and native access in neo4j," in *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, 2013, pp. 195–204.

[18] R. Angles, M. Arenas, P. Barceló, A. Hogan, J. Reutter, and D. Vrgoč, "Foundations of modern query languages for graph databases," *ACM Computing Surveys (CSUR)*, vol. 50, no. 5, pp. 1–40, 2017.