

بررسی و تحلیل تکثیر داده ها در پایگاه داده های توزیع شده

محمد رضا مهدوی مرام

دانشجوی کارشناسی ارشد، دانشکده مهندسی برق و کامپیوتر، دانشگاه آزاد اسلامی واحد آشتیان، ایران

چکیده

محبوبیت روزافزون تجزیه و تحلیل کلان داده و محاسبات ابری، استانداردهای مدیریت کلان داده جدیدی را ایجاد کرده است. تا به حال، پایگاه داده های رابطه ای یک انتخاب بهینه برای سازمان بودند. با این حال، با رشد مداوم داده های ذخیره شده و تحلیل شده، پایگاه های داده رابطه ای محدودیت های مختلفی را نشان می دهند، به عنوان مثال، محدودیتهای مقیاسپذیری و ذخیره سازی، و از دست دادن کارایی پرس و جو به دلیل حجم زیاد دادهها، و ذخیره و مدیریت پایگاههای داده بزرگتر چالش برانگیز میشوند. به منظور غلبه بر این محدودیت ها، یک مدل پایگاه داده جدید با مجموعه ای از ویژگی های جدید، به نام پایگاه های داده NoSQL توسعه یافت. پایگاه داده های غیر رابطه ای به عنوان یک فناوری پیشرفته ظاهر شدند و می توانند به تنهایی یا به عنوان مکمل پایگاه داده رابطه ای مورد استفاده قرار گیرند. یکی از چالش های مهم در این زمینه که تأثیر بسزایی در کارایی اینگونه سیستم ها دارد تکثیر داده ها است. فاکتور تکثیر بر در دسترس بودن داده ها تأثیر دارد که می تواند مربوط به زمان خواندن و در نهایت زمان اجرا باشد. انواع NoSQL زیادی با عملکردهای مختلف وجود دارد، بنابراین مقایسه آنها از نظر عملکرد مهم است. در این مقاله، ما سه پایگاه داده محبوب NoSQL را ارزیابی می کنیم: MongoDB، HBase، Cassandra. و به یک بررسی از الگوریتم های تکثیر برای سیستم های ذخیره سازی و مدیریت محتوای مختلف توزیع شده از جمله سیستم های مدیریت پایگاه داده توزیع شده، و آنالیز و بررسی و تکثیر داده ها در hbase، mongodb و cassandra پرداخته می شود. و سعی بر این است که عملکرد چندین سیستم توزیع شده را با استفاده از فاکتورهای تکرار آنالیز کرده و برای داده های مختلف ارزیابی کنیم و با مقایسه بر بهبود آن تأثیرگذار باشیم. فاکتور تکثیر در سیستم های پایگاه داده توزیع شده به منظور افزایش performance و افزایش دسترس پذیری^۱ استفاده میشود. در این خصوص هر سیستم توزیع شده ی پایگاه داده ی NoSQL سیاست های مخصوص خود را در راستای انتخاب نوع و تعداد replication ها انجام میدهد.

کلمات کلیدی: تکثیر داده ها، پایگاه داده های توزیع شده، پایگاه داده رابطه ای

¹ availability

Investigation and Analysis of Data Duplication in Distributed Database

Mohammad Reza Mahdavamaram

Faculty of Electrical and Computer Engineering, Islamic Azad University,
Ashtian Branch, Iran

Abstract

The growing popularity of big data analysis and cloud computing has created new big data management standards. Sometimes, programmers may interact with a number of heterogeneous data repositories, depending on the information they are responsible for. Until now, relational databases were an optimal choice for the organization. However, with the continuous growth of stored and analyzed data, relational databases show various limitations. For example: scalability and storage limitations, and loss of query performance due to large volumes of data, and storage and management of larger databases are challenging. In order to overcome these limitations, a new database model with a set of new features, called NoSQL databases, was developed. Non-relational databases emerged as an advanced technology and can be used alone or as a complement to relational databases. One of the important challenges in this field, which has a significant impact on the efficiency of such systems, is the reproduction of data. The replication factor affects data availability, which can be related to read time and ultimately execution time. There are many types of NoSQL with different functionality, so it is important to compare them in terms of performance and to examine how performance relates to the type of database. In this article, we evaluate three popular NoSQL databases: Cassandra, HBase, MongoDB, and a review of replication algorithms for various distributed storage and content management systems, including distributed database management systems, and analysis and review and replication of data in Mongoddb, Hbase, and Cassandra will be. It is an attempt to analyze the performance of several distributed systems using repetition factors and evaluate them for different data and influence its improvement by comparison. The replication factor is used in distributed database systems in order to increase performance and availability. Also, replication increases scalability in database systems. In this regard, each distributed NoSQL database system carries out its own policies in order to select the type and number of replications.

تاریخچه مقاله:

تاریخ ارسال: ۱۴۰۰/۰۳/۲۰

تاریخ اصلاحات: ۱۴۰۰/۰۶/۲۰

تاریخ پذیرش: ۱۴۰۰/۰۶/۳۰

تاریخ انتشار: ۱۴۰۰/۰۶/۳۱

Keywords:

Availability,
Data replication, Distributed
Databases, Relational
Databases,

ایمیل نویسنده مسئول:

mahdavamaram@gmail.com

۱ - مقدمه

سیستم پایگاهی توزیع شده^۲

پایگاه داده مجموعه ای است از داده های ذخیره شده و پایا، بصورت مجتمع (یکپارچه) (نه لزوماً همیشه بطور فیزیکی، بلکه حداقل بطور منطقی)، بهم مرتبط، حتی الامکان با کمترین افزونگی، (دارای یک ساختار منطقی مبتنی بر یک مدل داده ای و توصیف شده بویژه در محیط انتزاعی یعنی در چارچوب همان مدل داده ای)، تحت مدیریت یک سیستم کنترل متمرکز، مورد استفاده یک یا چند کاربر از یک (یا بیش از یک) «سیستم کاربردی»، بطور همزمان و اشتراکی [۱].

تکثیر از داده ها^۳ به معنای تهیه نسخه یا نسخه های پشتیبان از اطلاعات میزبانی شده روی یک NAS و انتقال آن ها به NAS دیگری است تا در صورت بروز خرابی اطلاعات اصلی، یک کپی دایمی از اطلاعات در دسترس باشد. مکان این ذخیره سازها می تواند متغیر باشد. به طور کلی سازمان ها از مکانیزم تکثیر داده ها برای تهیه نسخه پشتیبان لحظه ای یا غیر لحظه ای از اطلاعات استفاده می کنند. با این حال، نکته مهمی که باید به آن دقت کنید به کارگیری الگوی درست است. در فرایند تکثیر داده ضروری است تا ارزیابی های دقیقی انجام شود.

همان گونه که از نام این اصطلاح مشخص است، تکثیر داده ها به فرایند، کپی، همگام سازی، تهیه نسخه ای قابل اعتماد از اطلاعات و انتقال اطلاعات به مکان دیگری به غیر از محل ذخیره ساز اصلی اشاره دارد [۲].

۲- پیشینه پژوهش

الگوریتم های تکثیر پویا

تا به امروز الگوریتم های تکثیر متعددی مطرح شده است. ابتدا برخی از روش هایی که با استفاده از تاریخچه دسترسی ها به سایت سیستم توزیع شده است و الگوریتم های تکثیر داده خودکار شرح داده می شود و پس از آن برخی از الگوریتم های مبتنی بر واکنشی اولیه موجود بررسی خواهد شد.

۲-۱ - الگوریتم های تکثیر داده خودکار و سلسله مراتبی

این روش یک الگوریتم تکثیر پویا را ارائه می دهد که نسخه ها در آن بر اساس وزن ها ایجاد می شوند. امتیاز اصلی این الگوریتم، افزایش اهمیت رکوردهای دسترسی جدید با دادن وزن های بالاتر به آن ها می باشد و جایگزینی نسخه در سطح سایت است. در این روش فاصله زمانی

دسترسی ها از زمان کنون و تعداد دسترسی ها، عوامل اصلی تصمیم گیری در مورد محبوبیت فایل ها است [۶].

یک استراتژی سلسله مراتبی برحسب پهنای باند تعریف می شود که^۴ زمان دسترسی به داده را با جلوگیری از ازدحام در شبکه گرید داده، کاهش می دهد [۷]. در این مقاله هدف پیدا کردن یک توزیع مناسب از نسخه های تکرار در سطح شبکه است که ترافیک شبکه را کاهش می دهد. این مسئله به عنوان یک مسئله بهینه سازی مطرح می شود و از نوع NP-کامل می باشد. در ابتدا یک الگوریتم حریمانه و سپس یک الگوریتم ژنتیک برای توزیع نسخه ها در سطح شبکه توصیف می شود [۸].

این مقاله تکثیر را بر اساس وزن و هزینه نسخه ها بدست می آورد که به صورت خلاصه LWLC نامیده می شود [۹]. در فاز انتخاب نسخه، زمان انتقال نسخه ها محاسبه می شود تا نسخه ای که کمترین زمان انتقال را دارد برای تکثیر انتخاب شود. در فاز مکان یابی برای قرار دادن نسخه ها، مناسب ترین سایت برای قرارگیری نسخه ها انتخاب می شود. در فاز جایگزینی نسخه ها، هر یک از فایل ها دارای یک متغیر ارزش می باشند. ارزش نسخه توسط دو شاخصه وزن (که از تعداد دسترسی ها به آن فایل در بازه های زمانی متفاوت به دست می آید) و هزینه انتقال (که از نسبت اندازه فایل به پهنای باند در دسترس محاسبه می شود) تعیین می شود. اگر در سایت مورد نظر فضای کافی برای ایجاد نسخه جدید نباشد، فایل های با ارزش پایین در لیست حذف قرار می گیرند تا فضا برای تکثیر فایل جدید فراهم آید.

یک مکانیسم تکثیر داده پویا به نام آخرین دسترسی بزرگترین وزن^۵ پیشنهاد شده است. طراحی یک مدل بر اساس مدیریت تکثیر داده های متمرکز است. LALW^۵ یک فایل محبوب را برای تکثیر انتخاب کرده و تعداد کپی ها و سایت های مناسب تکثیر را محاسبه می کند وزن های مختلف برای هر رکورد دسترسی داده قرار گرفته است [۱۰]. رکوردهای دسترسی داده ها در گذشته وزن بالاتری را دارد. این موضوع نشان می دهد که رکوردها ارزش بالاتری نسبت به منابع دارند.

با استفاده از این استراتژی می توان از ویژگی محلی بودن در سطح شبکه بهره مند شد. استراتژی BHR با افزایش محلی بودن سطح شبکه، زمان دسترسی را پایین می آورد [۱۱].

⁴ Bandwidth Hierarchy Replication (BHR)

⁵ Latest Access Largest Weight

² - Distributed database system

³ Data Replication

استفاده از الگوریتم PSO جهت تکرار داده های محبوب در محیط پایگاه داده NoSQL: در این راستا باید توابع هزینه و نحوه جایگذاری مشخص شود.

گام سوم در روش اجرایی الگوریتم پیشنهادی ارائه یک تابع هزینه در راستای رسیدن به بهترین جمعیت در الگوریتم بهینه سازی ازدحام ذرات است. هدف از الگوریتم پیشنهادی PSO، کمینه کردن میزان هزینه برای تکرار داده های محبوب در پایگاه داده NoSQL است.

گام بعدی از روش پیشنهادی، استفاده از الگوریتم

جایگذاری در راستای مساله مکان یابی داده ها دارد. در این راستا جمعیت اولیه ایجاد شده و سپس با ارائه تابع برازندگی و استفاده از آن، بهترین ذره محاسبه خواهد شد.

ارزیابی روش ارائه شده: گام بعدی در اجرای روش پیشنهادی، پیاده سازی و ارزیابی روش پیشنهادی است. در این راستا از معیارهای متفاوتی از جمله هزینه و میزان ذخیره سازی اطلاعات در پایگاه داده NoSQL استفاده می شود.

۳-۲- پیکربندی پایگاه داده NoSQL و مشخص کردن عناصر و پارامترهای موجود

اولین گام از اجرای روش پیشنهادی، ایجاد پایگاه داده NoSQL و مشخص کردن عناصر و پارامترهای موجود در آن است. در این راستا، این عناصر عبارتند از:

M درایو با عناصر محاسباتی و ذخیره سازی مجزا در یک پایگاه NoSQL انتخابی با توپولوژی درختی مشخص شده اند.

Si نشان دهنده میزان ظرفیت ذخیره سازی درایو i است.

$C(I, j)$: به هزینه ارتباط بین دو درایو i و درایو j اشاره دارد. این هزینه در صورتی که بین این دو درایو لینک مستقیم باشد، برابر با هزینه خود این لینک بوده و در غیر این صورت هزینه ارتباط دو درایو برابر مجموع هزینه های ارتباط غیرمستقیم دو درایو است. همچنین شایان ذکر است که هزینه لینک از درایو i به درایو j برابر با هزینه لینک از درایو j به درایو i است.

N به تعداد فایل ها اشاره دارد که با نام های O1, O2... ON و حجم های o1, o2, ..., ok در پایگاه داده NoSQL شناخته خواهند شد.

۳-۳- استفاده از الگوریتم PSO جهت تکرار داده های محبوب

با فرض ویژگی محلی بودن در فرآیند مکان یابی نسخه ها، یک الگوریتم مکان یابی مقیاس پذیر بر اساس الگوریتم chord تعریف می شود [۱۲].

توپولوژی درختی با توپولوژی حلقه ای ساختار نظیر به نظیر ترکیب شده است تا توپولوژی شبکه تکثیر را مدل کند. نتایج حاصل از شبیه سازی حاکی از این است که کارایی بهتری در درخواست و بهنگام سازی وجود دارد. اگر نسخه ها را بر اساس جدول مسیریابی در سایت های همسایه قرارداد. ایده اصلی این الگوریتم این است که یک کپی در نزدیک ترین سایت ممکن که این فایل را درخواست کرده، با نرخ بالاتر از حد آستانه تعریف شده ایجاد می شود. [۱۳].

۲-۲- الگوریتم های مبتنی بر واکنشی اولیه

یک استراتژی پویایی ارائه می شود که از الگوریتم فازی بهره می برد. این استراتژی در دو فاز اجرا می شود: ۱. درخواست و ایجاد نسخه جدید ۲. جایگزینی [۱۸]. در فاز اول اگر در حافظه فضای کافی وجود داشته باشد تکثیر انجام می شود، اما اگر فضای کافی موجود نبود، هزینه دسترسی به فایل به صورت راه دور محاسبه می شود و در صورتی که از یک حد آستانه بیشتر باشد، تکثیر باید اجرا شود؛ اگر هزینه از حد آستانه کمتر باشد تعداد نسخه ها بررسی می شود و اگر تعداد نسخه ها از یک حد آستانه بیشتر باشد، نیازی به نسخه جدید وجود ندارد و گرنه نسخه جدید باشد در حافظه ایجاد شود.

در فاز دوم با استفاده از نظریه فازی، دو معیار تعداد دسترسی ها و هزینه انتقال به الگوریتم فازی داده شده است و خروجی حاصل به عنوان ارزش نسخه ها در نظر گرفته شده است. نتایج حاصل از شبیه سازی نشان می دهد که این الگوریتم زمان اجرا و کارایی را بهبود می بخشد [۱۹].

۳- روش پیشنهادی

۳-۱- گام های اجرایی روش پیشنهادی

با توجه به توضیحات گفته شده، گام های اجرایی روش پیشنهادی به صورت موارد زیر شمرده می شود:

مشخص کردن عناصر و پارامترهای موجود در محیط پایگاه داده NoSQL: اولین گام در راستای انجام روش پیشنهادی، مشخص کردن عناصر و پارامترهای موجود در محیط پایگاه داده NoSQL است. از جمله این عناصر و پارامترها می توان به تعداد درایوها، توپولوژی انتخابی، میزان ظرفیت ذخیره سازی درایوها و ... اشاره کرد.

است، تعداد درخواست‌ها برای درایو است که در واقع درایو با تعداد درخواست بیشتر، شانس بیشتری برای انتخاب شدن خواهد داشت. هزینه ذخیره‌سازی (SC): دومین فاکتوری که برای محاسبه میزان برآزندی هر ذره مورد اهمیت قرار گرفته است، هزینه ذخیره‌سازی است. در این تحقیق، هزینه ذخیره‌سازی برابر با فضای استفاده شده برای ذخیره داده است و بصورت (رابطه-۲) محاسبه شده است:

$$SC = \frac{\text{file.size}}{\text{Available space of site}_i} \quad (\text{رابطه-۲})$$

که در (رابطه -۳)، File.size به اندازه فایل اشاره دارد و available space of site_i نیز به میزان فضای آزاد موجود در درایو i اشاره دارد.

هزینه انتقال (TC): سومین فاکتور برای محاسبه میزان برآزندی، هزینه انتقال است و برابر با هزینه انتقال فایل داده‌ای از درایو دارنده نسخه تا درایو درخواست‌دهنده است. درایوی از درجه اهمیت بیشتری برخوردار است که میانگین هزینه انتقال آن کمتر باشد. برای این فاکتور از رابطه-۳ استفاده می‌شود:

$$TC = \frac{\text{file.size}}{\text{Bandwidth}_{si,sj}} \quad (\text{رابطه-۳})$$

که S_i درایو مورد نظر برای قرار دادن تکرار و S_j درایو غیر تکرار است. سپس برای محاسبه هزینه انتقال کل نیز از میانگین هزینه انتقال استفاده می‌شود که بصورت (رابطه-۴) نشان داده شده است:

$$AvgTC = \frac{\sum_{j=1}^{m-1} \text{file.size}}{m-1} \quad (\text{رابطه-۴})$$

که در این رابطه m تعداد درایوهای غیر تکرار را نشان داده و Bandwidth نیز پهنای باند میان درایو i و درایو j را نشان می‌دهد. درایوهای تکرار: چهارمین فاکتور برای محاسبه میزان برآزندی، درایوهای تکرار هستند. درایوهای تکرار هرگز فایلی را که به صورت محلی در اختیار دارند، درخواست نمی‌کنند. از این رو، درایوها میزبان جدید باید تا حد امکان از درایوهای تکرار دور باشند تا بار کاری توزیع شود. میانگین پهنای باند میان این درایوها، از رابطه بعدی محاسبه می‌شود:

پس از پیکربندی پایگاه داده NoSQL و مقداردهی پارامترها و عناصر موجود در این محیط، در مرحله بعدی باید متدی جهت تکثیر داده‌های محبوب در پایگاه NoSQL ارائه دهیم. در این راستا در این تحقیق از الگوریتم PSO استفاده می‌شود.

روند اجرایی الگوریتم PSO در راستای انتخاب فایل‌های محبوب، به این صورت است که در ابتدا جمعیت اولیه برای الگوریتم PSO ایجاد می‌شود. هر جمعیت در این الگوریتم، بصورت (شکل-۱) در نظر گرفته می‌شود:

Particle1	Particle2	Particle3	...	Particlen
-----------	-----------	-----------	-----	-----------

(شکل-۱) اعضای هر جمعیت در روش پیشنهادی

همانطوری که از (شکل-۱) مشخص شده است، هر جمعیت از تعدادی ذره تشکیل شده است که خود هر ذره نیز به نوبه خود، بصورت (شکل-۲) نشان داده شده است:

Drive1	Drive2	Drive3	...	Driven
--------	--------	--------	-----	--------

(شکل-۲) اعضای تشکیل‌دهنده هر ذره در روش پیشنهادی

همانطوری که مشخص شده است، هر ذره در روش پیشنهادی، از تعدادی درایو تشکیل شده است. به عبارت دیگر، اگر مثلاً فرض کنیم که از ما ۵ درایو محبوب را از بین ۱۰۰ درایو موجود بخواهند، باید تمام حالات ۵ عضوی از این مجموعه ۱۰۰ عضوی را در نظر بگیریم. یعنی در واقع باید ترکیب ۵ از ۱۰۰ را محاسبه کرده که تمام حالات زیرمجموعه ۵ عضوی را نشان می‌دهد. در مرحله اول از الگوریتم پیشنهادی PSO، جمعیت بصورت تصادفی ایجاد می‌شود. یعنی براساس (رابطه-۱) داریم:

$$\text{Population} = \{\text{Generate Random } n \text{ Particle}\} \quad (\text{رابطه-۱})$$

که براساس (رابطه-۱)، n ذره بصورت تصادفی در روش پیشنهادی تولید می‌شود که ذره مطابق با (شکل-۲)، از تعدادی درایو تشکیل شده است.

حال در مرحله بعدی باید برای هر ذره در روش پیشنهادی، تابع برآزندی فراخوانی شده و میزان برآزش مورد محاسبه قرار بگیرد. در این راستا، فاکتورهایی که برای محاسبه میزان برآزندی مورد استفاده قرار می‌گیرند، بصورت زیر ارائه شده‌اند:

تعداد درخواست‌های هر درایو غیر تکرار برای فایل مورد نظر: اولین فاکتوری که برای محاسبه میزان برآزندی مورد اهمیت قرار گرفته

```

For(p=1;p<=Swarm.size;p++){
    Generate Particle Randomly Included m from
    n Sites.
    Generate X and V Randomly for Each Particle.
}
Do{
    For each Particle Xp{
        Calculate Fitness Value of Xp Using equation
        3-7.
        If the fitness value is better than the best
        fitness value in
            History then set current value as the New
            Pbest.
    }
    Choose the Particle with the best fitness value as the
    gbest;
    For each Particle Xp{
        Calculate Particle Velocity using equation 3-
        8.
        Update Particle Position according equation
    } While (one of the termination condition is not
    satisfied)
    Return best Particle;
}

```

۳-۴- جایگزینی

پس از انتخاب بهترین درایوها برای جایگذاری، نسخه‌های جدید باید در این درایوها ذخیره شوند. در این بخش، از پژوهش، فضای ذخیره‌سازی هریک از درایوهای انتخاب شده بررسی می‌شوند. این بررسی منجر به ۳ حالت زیر می‌شود:

کل فضای ذخیره‌سازی درایوها از اندازه فایل کمتر باشد.

در درایو میزبان، فضای آزاد کافی برای عمل تکثیر وجود داشته باشد.

ذخیره‌سازی فایل در گره مقصد امکان‌پذیر باشد ولی فضای آزاد کافی برای عمل تکثیر وجود نداشته باشد. در این حالت، باید فایل‌هایی که ارزش کمتری دارند از درایو مقصد حذف شوند. ارزش‌گذاری هریک از فایل‌ها با توجه به ۳ معیار تعداد فایل‌های موجود از فایل در شبکه، تعداد دفعات دسترسی به فایل در گذشته و آخرین زمان دسترسی به فایل بدست می‌آید که هریک از این معیارها در ادامه بررسی شده‌اند. این معیارها عبارتند از: تعداد نسخه‌های موجود از فایل در شبکه (NOR) فایلی با ارزش‌تر است که تعداد نسخه‌های کمتری از آن در شبکه موجود باشد. تعداد دفعات دسترسی به فایل در گذشته (NRR)

$$Avg_{Bandwidth} = \frac{\sum_{j=1}^{m-1} Bandwidth_{si,sj}}{m-1} \quad \text{(رابطه-۵)}$$

حال برای محاسبه میزان هزینه کل، از رابطه‌های زیر استفاده می‌شود:

$$Total - Cost(Si) = Avg_{TC}(Si) + SC(Si) + \frac{1}{Avg_{Bandwidth}} + \frac{1}{\#Request} \quad \text{(رابطه-۶)}$$

که نحوه محاسبه تمامی فاکتورهای موجود در رابطه-۶، در قسمت‌های قبلی بطور کامل بیان شده است. همچنین برای محاسبه میزان برازندگی هر ذره از رابطه بعدی استفاده شده است:

$$Fitness(k) = \frac{1}{\sum_{i=1}^n Total - Cost(Si)} \quad \text{(رابطه-۷)}$$

براساس رابطه-۷ که در راستای محاسبه میزان برازندگی هر ذره استفاده شده است، هرچه میزان هزینه بیشتر باشد، حال پس از محاسبه میزان برازندگی، باید ذره موردنظر جایجا شود. برای این کار از دو مولفه سرعت و مکان استفاده شده است. سرعت هر ذره مطابق با رابطه-۸ محاسبه می‌شود:

$$V[] = V[] + c1 * rand() * (Pbest[] - position[]) + c2 * rand() * (Gbest - position[]) \quad \text{(رابطه-۸)}$$

که در رابطه-۸، Pbest به بهترین موقعیت ذره تاکنون و Gbest نیز به بهترین موقعیت اشاره دارد.

همچنین برای بروزرسانی مکان ذره نیز از رابطه-۹ استفاده می‌شود:

$$Position[] = Position[] + V[] \quad \text{(رابطه-۹)}$$

براساس توضیحات ارائه شده برای الگوریتم پیشنهادی PSO، شبه‌کد پیشنهادی زیر، روند انجام این الگوریتم را بهتر نشان داده است:

Particle PSO()
{

مقدار هزینه بدست آمده: یکی از پارامترهایی که برای تجزیه و تحلیل الگوریتم پیشنهادی در پایگاه داده NoSQL مورد استفاده قرار می‌گیرد، مقدار هزینه و یا همان برانزنگی ذرات در الگوریتم پیشنهادی می‌باشد که می‌توان برحسب پارامترهای مختلفی از جمله تعداد سرورها، مورد بررسی قرار گیرد. مقدار داده ذخیره شده: یکی دیگر از معیارهای کارایی که می‌توان برای روش پیشنهادی استفاده کرد، مقدار داده ذخیره شده است.

۴- سیستم پیاده‌سازی

با توجه به گام‌های اجرایی روش پیشنهادی، برای پیاده‌سازی این روش، سیستم باید دارای حداقل امکاناتی باشند که این مشخصات در جدول-۱ نشان داده شده است:

(جدول ۱- مشخصات سیستم برای پیاده‌سازی روش پیشنهادی)

مقدار	معیار
Core i7 با سرعت ۶.۷۳ گیگاهرتز	سرعت پردازنده
۶ گیگابایت	میزان حافظه داخلی Ram
۴ گیگابایت	حافظه Cache
ویندوز ۱۰ نسخه ۶۴ بیتی	سیستم عامل
Matlab نسخه ۲۰۱۶	نرم‌افزار پیاده‌سازی

برای پیاده‌سازی الگوریتم پیشنهادی از نرم‌افزار پیاده‌سازی Matlab نسخه ۲۰۱۶ استفاده شده است.

راه‌اندازی پایگاه داده NoSQL و مقداردهی اولیه

پارامترهای آن اولین گام از روش پیشنهادی، راه‌اندازی پایگاه داده NoSQL و در راستای آن، مشخص کردن پارامترها و مقداردهی اولیه این پارامترها است. جدول ۲-۴، پارامترهای لازم برای راه‌اندازی پایگاه داده NoSQL و همچنین مقداردهی اولیه آن‌ها را نشان داده است:

(جدول ۲- پارامترها و مقداردهی اولیه آن‌ها برای راه‌اندازی پایگاه داده NoSQL)

فایلی که در گذشته تعداد دفعات دسترسی بیشتری به آن انجام شده باشد، فایل با ارزش تری خواهد بود. مقدار این پارامتر با محاسبه است:

$$NRR = \frac{Request_{fi}}{T_{Current} - T_{store}} \quad \text{(رابطه ۱۰)}$$

که در آن Request تعداد دفعات درخواست‌ها برای فایل و T(Current) زمان جاری و T(store) زمان ذخیره فایل در یک گره خاص است.

آخرین زمان دسترسی به فایل (LAT) هرچه زمان دسترسی به فایل نزدیکتر باشد و یا به عبارت دیگر، هرچه فاصله زمانی میان زمان جاری و آخرین زمان دسترسی به یک فایل کمتر باشد، آن فایل با ارزش تر است. این زمان از طریق رابطه-۱۱ محاسبه می‌شود:

$$LAT = T_{Current} - T_{Last\ access\ time} \quad \text{(رابطه ۱۱)}$$

با در نظر گرفتن این معیارها، برای محاسبه ارزش هریک از فایل‌های موجود در درایو مقصد، از رابطه ۱۲ استفاده می‌شود:

(رابطه-۱۲)

$$Rep\ Val(RV) = w1 * \frac{1}{LAT} + w2 * NRR + w3 * \frac{1}{NOR}$$

که در این رابطه ضرایب $w1$ تا $w3$ ضرایب اهمیت هریک از معیارها را نشان داده است که در اینجا به ترتیب برابر با ۰.۳ و ۰.۴ و ۰.۳ در نظر گرفته شده است.

در پایان کار، الگوریتم فایلها را به صورت صعودی با توجه به ارزششان مرتب کرده و از ابتدای لیست تا زمانی که فضای کافی برای قرار دادن فایل‌های جدید در گره مقصد به وجود آید، حذف می‌کند.

۴-۳- استراخ ویژگی ارزیابی متد پیشنهادی

پس از آنکه با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات پرندگان، اقدام به تشخیص فایل‌های محبوب برای تکثیر در محیط پایگاه داده NoSQL شد، در مرحله بعدی باید متد پیشنهادی را بر روی یک مطالعه موردی، پیاده‌سازی کرده و نتایج بدست آمده را مورد ارزیابی و تجزیه و تحلیل قرار داد. در این تحقیق برای ارزیابی متد پیشنهادی از پارامترهای زیر استفاده خواهد شد:

برازندگی پیشنهادی، کاملاً منطقی به نظر می‌رسد. همچنین (شکل-۴)، نمایی از خروجی روش پیشنهادی را نشان داده است:

```

Iter = 78 BEST = 0.28307
Iter = 79 BEST = 0.28307
Iter = 80 BEST = 0.28307
Iter = 81 BEST = 0.28307
Iter = 82 BEST = 0.28307
Iter = 83 BEST = 0.28307
Iter = 84 BEST = 0.28307
Iter = 85 BEST = 0.28307
Iter = 86 BEST = 0.28307
Iter = 87 BEST = 0.28307
Iter = 88 BEST = 0.28307
Iter = 89 BEST = 0.28307
Iter = 90 BEST = 0.28307
Iter = 91 BEST = 0.28307
Iter = 92 BEST = 0.28307
Iter = 93 BEST = 0.28307
Iter = 94 BEST = 0.28307
Iter = 95 BEST = 0.28307
Iter = 96 BEST = 0.28307
Iter = 97 BEST = 0.28307
Iter = 98 BEST = 0.28307
Iter = 99 BEST = 0.28307
Iter = 100 BEST = 0.28307
-----
Sites is Selected is =
Columns 1 through 8
    7    11    14    22    23    25    26    27
Columns 9 through 16
   29    31    41    43    44    49    63    66
Columns 17 through 21
   72    78    84    94    100
    
```

(شکل-۵) نمایی از خروجی روش پیشنهادی

همانطوری که از شکل ۵ مشخص شده است، تکرارهای ۷۸ تا ۱۰۰ با مقادیر بهترین برازندگی در برابر تکرار نشان داده شده است. همچنین در انتها، بهترین زیرمجموعه‌ای از سایت‌ها که به عنوان سایت‌های محبوب انتخاب شده‌اند، نشان داده شده است.

۵- نتیجه گیری

نتایج خروجی روش پیشنهادی

مرحله بعدی از این تحقیق، دادن ورودی‌های مختلف و رسم نمودارهای متناظر با این ورودی‌ها است. این ورودی‌ها عبارتند از:

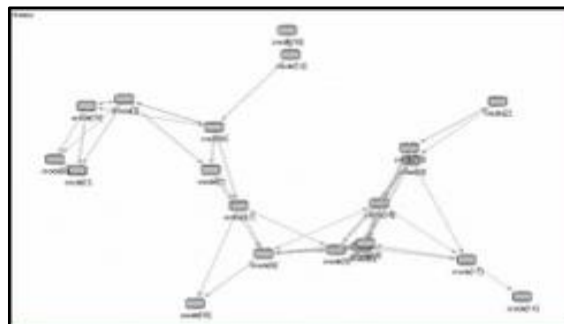
- سناریو اول: تعداد سایت‌ها متغیر باشد.
- سناریو دوم: تعداد فایل‌ها متغیر باشد.

براساس سناریوهای گفته شده، در هر بار، هر کدام از این متغیرها را تغییر داده و معیار درصد حجم باقیمانده سایت‌ها محاسبه می‌شود. جدول ۳، معیار حجم باقیمانده سایت‌ها را براساس تعداد متغیر سایت‌ها، نشان داده است:

(جدول-۲) پیاده‌سازی روش پیشنهادی

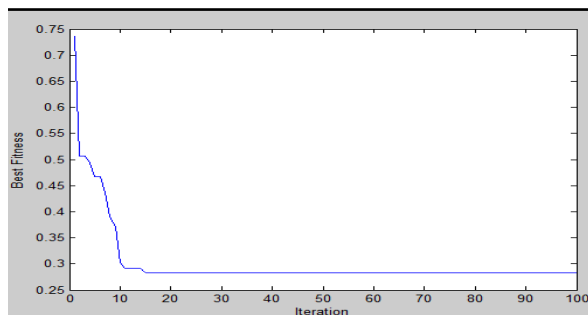
پارامترها	مقدار
تعداد سایت	۴
ظرفیت ذخیره‌سازی	۲ کیلوبایت تا ۷ مگابایت
هزینه ارتباط	۱ تا ۱۰
تعداد فایل	۱۰۰
تعداد درخواست	بین ۲ تا ۱۰
ظرفیت هر سایت	بین ۱۰۰ کیلوبایت تا ۱۰۰ مگابایت

مرحله اول پیاده‌سازی، نمایی از ساختار پایگاه داده NoSQL مورد تحقیق است. شکل ۴-۲، نمایی از ساختار ارتباطی پایگاه داده NoSQL پیشنهادی را نشان داده است:



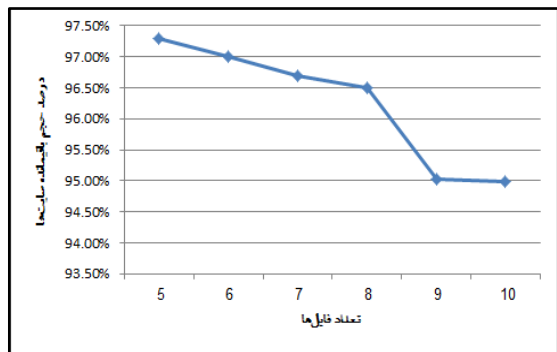
(شکل-۳) نمایی از محیط و پایگاه داده NoSQL مورد نظر

مقدار بهترین برازندگی را در برابر تکرار نشان داده است:



(شکل-۴) مقدار بهترین برازندگی در برابر تکرار

تابع برازندگی پیشنهادی متشکل از ۴ پارامتر زمان، هزینه، تعداد درخواست و پهنای باند است که با توجه به این پارامترها، هرچه میزان برازندگی کمتر باشد، ذره انتخاب شده، از درجه اهمیت بالاتری برخوردار بوده و بهینه‌تر است که این موضوع با توجه به ماهیت الگوریتم تکاملی بهینه‌سازی ذرات و همچنین تابع



شکل ۶- درصد حجم باقیمانده سایت براساس تعداد متغیر

فایل‌ها

همانطوری که از شکل ۶- مشخص شده است، با افزایش تعداد فایل‌ها، حجم باقیمانده سایت‌ها کاهش پیدا کرده است

۶- نتیجه‌گیری و پژوهش‌های آتی

در مرحله بعدی این تحقیق، باید نتایج خروجی روش پیشنهادی را با سایر روش‌های دیگر مورد بررسی و تجزیه و تحلیل قرار دهیم. در این راستا، روش‌های بسیاری ارائه شده‌اند، ولیکن روش‌هایی که برای مقایسه روش پیشنهادی در این تحقیق در نظر گرفته شده‌اند، به صورت موارد زیر برشمرده شده‌اند:

الگوریتم شبیه‌سازی تیریدی.

۲- الگوریتم ژنتیک.

اولین مرحله از انجام عملیات مقایسه، تعیین پارامترهای ورودی مطابق با پیاده‌سازی روش‌های ذکر شده است.

جدول ۵- پارامترها و مقادیر اولیه آن‌ها برای مقایسه

پارامترها	مقدار
تعداد سایت	بین ۱۰ تا ۱۵۰
ظرفیت ذخیره‌سازی	۲ کیلوبایت تا ۷ مگابایت
هزینه ارتباط	۱ تا ۱۰
تعداد فایل	۱۰۰
تعداد درخواست	بین ۲ تا ۱۰
ظرفیت هر سایت	بین ۱۰۰ کیلوبایت تا ۱۰۰ مگابایت

جدول ۳- حجم باقیمانده سایت‌ها براساس تعداد متغیر سایت‌ها

تعداد سایت	حجم باقیمانده سایت
۶۰	۹۷/۸٪
۷۰	۹۸/۸٪
۸۰	۹۸/۵٪
۹۰	۹۸/۲٪
۱۰۰	۹۸/۲٪
۱۱۰	۹۸/۱۸٪
۱۲۰	۹۸/۱۵٪
۱۳۰	۹۸/۱۱٪
۱۴۰	۹۸/۱٪
۱۵۰	۹۷/۵٪

با افزایش تعداد سایت‌ها، حجم باقیمانده سایت‌ها کاهش پیدا کرده است و این موضوع با توجه به تعداد درخواست‌های سایت‌های اضافه شده، کاملاً منطقی به نظر می‌رسد.

در سناریوی دوم، تعداد فایل‌ها را برابر با متغیر و تعداد سایت‌ها را برابر با ۱۰۰ در نظر گرفته و نتایج را مورد تجزیه و تحلیل قرار می‌گیرد. جدول ۴-۴، معیار حجم باقیمانده سایت‌ها را براساس تعداد متغیر فایل‌ها، نشان داده است:

جدول ۴- حجم باقیمانده سایت‌ها براساس تعداد متغیر فایل‌ها

تعداد سایت	حجم باقیمانده سایت
۵	۹۷/۳٪
۶	۹۷٪
۷	۹۶/۷٪
۸	۹۶/۵٪
۹	۹۵/۰۳٪
۱۰	۹۴/۹۹۵٪

همچنین شکل ۶، نمودار درصد حجم باقیمانده سایت را براساس تعداد متغیر فایل‌ها، نشان داده است:

گفته شده یعنی ژنتیک و شبیه سازی تیریدی، بهتر عمل کرده است. در حالت تعداد سایت برابر با ۶۰، از الگوریتم ژنتیک بدتر عمل کرده است و در حالت تعداد سایت برابر با ۸۰ نیز با الگوریتم ژنتیک مساوی عمل کرده است.

پس از مقداردهی اولیه پارامترها، جدول ۶-، نتایج خروجی را نشان داده است:

(جدول ۶- مقایسه درصد حجم باقیمانده سایتها براساس تعداد متغیر سایتها)

تعداد سایت	۱۰	۲۰	۴۰	۶۰	۸۰	۱۰۰	۱۵۰
SA	۴۳	۴۱	۵۰	۷۰	۶۸	۷۰	۶۸
GA	۴۱	۴۰	۴۲	۳۸	۵۰	۵۵	۶۰
Proposed Method	۳۶	۳۸	۴۰	۴۹	۵۰	۵۲	۵۷

۷- مراجع

[1]. Khashan E, Eldesouky A, Elghamrawy S, An adaptive spark-based framework for querying large-scale NoSQL and relational databases- (2021) An adaptive spark-based framework for querying large-scale NoSQL and relational databases. PLoS ONE 16(8): e0255562.

[2]. Abramova v, Bernardino j, Furtado p, Which NoSQL Database? A Performance Overview - Open Journal of Databases (OJDB) Volume 1, Issue 2, 2014 by the authors; licensee RonPub, Lübeck, Germany - ISSN 2199-3459

[3].sareen p , kumar p , NOSQL DATABASE AND ITS COMPARISON WITH SQL database- International Journal of Computer Science & Communication Networks, Vol 5(5),293-298 - ISSN:2249-5789- 2014

[4]. Haughian g, Osman r, knottenbelt w, Benchmarking Replication in Cassandra and MongoDB NoSQL Datastores - September 2016 - DOI:10.1007/978-3-319-44406-2_12 -Conference: International Conference on Database and Expert Systems Applications

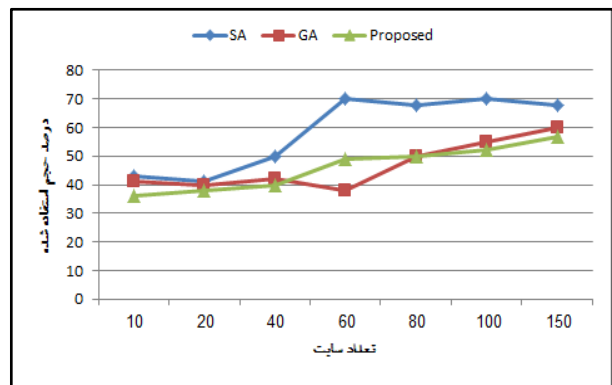
[5]. H.egemen, l.batista, e.cuaha, Investigation of Replication Factor for Performance Enhancement in the Hadoop Distributed File System - Companion of the 2018 ACM/SPEC International Conference - At: Berlin - DOI:10.1145/3185768.3186359 - April 2018

[6]. K.kituta.ezechiel, sh.kant, r.agarwal , Analysis of database replication protocols- international Journal of Latest Trends in Engineering and Technology, Special Issue ICRMR-2018, pp. 075-083, e-ISSN:2278-621X

[7]. S-Yih Hwang, K- K. S. Lee2 , Y. H. Chin, Data Replication in a Distributed System: A Performance Study, Chapter · October 2006- DOI: 10.1007/BFb0034724 · Source: CiteSeer

[8]. T. Lakshmi Siva Rama Krishna, J. Priyanka, N. Nikhil Teja, Sd. Mahiya Sultana, B. Jabber - An Efficient Data Replication Scheme for Hadoop Distributed File System - International Journal of Engineering & Technology, 7 (2.32) (2018) 167-169

همچنین شکل-۷، مقایسه نتایج خروجی را برای روش پیشنهادی و سایر روش های گفته شده، نشان داده است:



(شکل ۷- مقایسه حجم استفاده شده در سایتها برای روش های ارائه شده)

همانطوری که از شکل-۷ مشخص شده است، درصد حجم استفاده شده برای روش پیشنهادی در اکثر حالات نسبت به سایر روش های گفته شده یعنی شبیه سازی تیریدی و الگوریتم ژنتیک، بهتر بوده و لذا از این نقطه نظر بهینه است. دلیل این بهینگی این است که با توجه به اینکه در روش پیشنهادی، از ۴ پارامتر هزینه، تعداد درخواست، زمان و پهنای باند استفاده شده است، لذا الگوریتم پیشنهادی مبتنی بر بهینه سازی ذرات پرندگان PSO، قصد دارد که این تابع برازندگی را بهینه کند. لذا براساس این توضیحات، نتایج حاصل از روش پیشنهادی یعنی حجم استفاده شده توسط سایتها با توجه به افزایش تعداد سایتها، افزایش پیدا کرده است ولی این افزایش در اکثر حالات یعنی همه حالات به جز تعداد سایت برابر با ۶۰ و ۸۰، نسبت به دو روش



محمد رضا مهدوی مرام فوق دیپلم کامپیوتر
گرایش نرم افزار را از دانشکده شهید شمسی
پور تهران در سال ۱۳۸۳، کارشناسی مهندس
کامپیوتر گرایش نرم افزار را از دانشگاه پیام نور

قم در سال ۱۳۹۲ اخذ کردند و هم اکنون کارشناسی ارشد مهندس
فناوری اطلاعات گرایش شبکه های کامپیوتری در دانشگاه آزاد اسلامی
واحد آشتیان در حال اخذ می باشد. ایشان از سال ۱۳۹۲ تاکنون مدیر
فناوری اطلاعات و ارتباطات بانک انصار استان قم میباشند.

نشانه رایانامه ایشان عبارتند از:

mahdavimaram@gmail.com

روش ارجاع: م. مهدوی. بررسی و تحلیل تکثیر داده ها در پایگاه داده
های توزیع شده (NoSQL)، دوفصلنامه محاسبات و سامانه
های توزیع شده، سال چهارم، شماره اول، ص ۱۰۱-۱۱۱، سال
۱۴۰۰

cite: Mohammad Reza Mahdavi Maram, Investigation and analysis of replication in distributed database (NoSQL), Journal of Distributed Computing and Systems (JDCCS), Vol ۴, Issue ۱, Page 101-111.

[9]. kuppusamy, p.elango - Data Replication for the Distributed Database using Decision Support Systems- International Journal of Computer Applications (0975 – 8887) - Volume 69– No.3, May 2013

[10]. B.ciciani, d.dias, - Analysis of Replication in Distributed Database Systems - IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 2 . NO. 2 . JUNE 1990 , IEEE, AND PHILIP s. Yu, SENIOR MEMBER, IEEE

[11]. . Sun, J. Zheng, Q. Liu and Y. Liu, "Dynamic Data Replication Based on Access Cost in Distributed Systems," 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology, 2009, pp. 829-834, doi: 10.1109/ICCIT.2009.198

[12]. G.wang , j.koshy, s.sabramanian- Building a replicated logging system with Apache Kafka Share on- Proceedings of the VLDB Endowment Volume 8 Issue 12 August 2015 pp 1654–1655

[۱۳]. ایزنگنه، م. مرادی، م. مهدوی - مقایسه تکثیر داده ها در سیستم

های توزیع شده - اولین همایش منطقه ای رویکرد هلی نوین در مهندسی کامپیوتر و فناوری اطلاعات - ۵ خرداد ۱۳۹۰ - رودسر

[۱۴]. م. نصرتی، م. فضلعلی، تکثیر منابع داده در سیستم توزیع شده بر اساس روش تشخیص جوامع، اولین همایش ملی نگرشی نوین در مهندسی برق و کامپیوتر اسفند ۱۳۹۵ - دانشگاه آزاد اسلامی واحد کرمانشاه، گروه برق و کامپیوتر -

[۱۵]. ف. پدیداران مقدم، ف. رضایی - بررسی الگوریتم های زمانبندی در سیستم های توزیع شده - تحقیقات نوین در زمینه برق و کامپیوتر و فناوری اطلاعات، دانشگاه آزاد اسلامی واحد مبارکه - - ecit 2018,10

[16]. <http://www.mongodb.org/about/introduction/>

[17]. <http://wiki.apache.org/cassandra/>