ارائه راهکاری نوین برای کشف تغییرات بارکاری در پایگاه‌داده خودتنظیم NoSQL

مریم مظفری[۱]، دانشجوی دکتری؛ اسلام ناظمی[۲]، دانشیار؛ امیرمسعود افتخاری مقدم[۳]، دانشیار

۱- دانشکده مهندسی کامپیوتر و فناوری اطلاعات - واحد قزوین، دانشگاه آزاد اسلامی- قزوین- ایران – m_mozaffari@qiau.ac.ir
۲- دانشکده مهندسی و علوم کامپیوتر - دانشگاه شهید بهشتی - تهران - ایران - nazemi@sbu.ac.ir
۳- دانشکده مهندسی کامپیوتر و فناوری اطلاعات - واحد قزوین - دانشگاه آزاد اسلامی – قزوین - ایران - eftekhari@qiau.ac.ir

چکیده: سیستم‌های مدیریت پایگاه داده قسمت اصلی سیستم‌های اطلاعاتی هستند که اندازه و پیچیدگی این سیستم‌ها به‌طور چشمگیری در سال‌های اخیر افزایش یافته است. با رشد و پیشرفت سیستم‌های پایگاه داده و پیچیده‌تر شدن آن‌ها، مدیران پایگاه داده با مشکلات و چالش‌های بیشتری روبرو شده‌اند و مدیریت این سیستم‌ها پرهزینه و زمان‌بر است. قسمت اصلی هزینه کلی مالکیت سیستم پایگاه داده، شامل هزینه‌های مدیر خبره‌ای است که بتواند این سیستم‌های بزرگ و پیچیده را مدیریت کند. پایگاه داده خودمختار با فراهم آوردن عملکرد خود مدیریتی، منجر به کاهش هزینه کل مالکیت برای سیستم پایگاه داده می‌شود. تصمیمات خود مدیریتی نظیر تنظیم خودکار شِمای پایگاه داده وابسته به بارکاری پایگاه داده است. بنابراین یکی از مسائل مهم در تحقق تنظیم خودکار پایگاه داده، پایش و تحلیل بارکاری برای کشف تغییرات و تطبیق طراحی (بازتنظیمی) شِمای پایگاه داده با این تغییرات است. در این مقاله یک حلقه کنترل بازخورد برای پایش پیوسته و تحلیل سبک وزن بارکاری در پایگاه داده ستون‌گرا NoSQL ارائه می‌شود. این حلقه توصیف کننده یک الگوی طراحی برای ویژگی خودتنظیمی است و برای کشف تغییرات بارکاری بکار می‌رود که لازمه بازتنظیمی خودکار شِمای پایگاه داده است. نتایج حاصل شده از آزمایش‌ها کارایی راهکار پیشنهادی در کشف تغییرات بارکاری را نشان می‌دهد.

واژه‌های کلیدی: پایگاه داده خودمختار، تنظیم خودکار شِمای پایگاه داده، پایش و تحلیل بارکاری، حلقه کنترل بازخورد، پایگاه داده NoSQL.

A New Solution for Workload Change Detection in Self-Tuning NoSQL Database

M. Mozaffari[1], PhD Student; E. Nazemi[2], Associate Professor; A. M. Eftekhari-Moghadam3, Associate Professor

1- Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran, Email: m_mozaffari@qiau.ac.ir
2- Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran, Email: nazemi@sbu.ac.ir
3- Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran, Email: eftekhari@qiau.ac.ir

Abstract: Database management systems are the main part of information system that the size and complexity of these systems significantly have been increased in recent years. With the growing and being more complicated database management systems, database Administrators face more problems and challenges, so management of these systems is Time-consuming and costly. More over the main part of the total cost of ownership includes the cost of expert database administrator (DBA) who can manage these large and complicated systems. Autonomic databases by providing self-management functionality are caused to reduce the total cost of ownership for a database system. The self-management decisions as the automated schema database tuning depend on the database workload. One of the important issues in realizing the database automated tuning is workload monitoring and analysis for changes detection and schema re-tuning with this changes. In this paper is presented the feedback control loop for continuous monitoring and light-weight analysis of workload in NoSQL column-oriented database. This loop describes a design pattern for self-tuning feature and uses for workload change detection which require automated schema database re-tuning. The experimental results exhibit the effectiveness of the proposed solution for workload change detection.

Keywords: Autonomic databases, Automated schema database, Workload monitoring and analysis, Feedback control loop, NoSQL database.

تاریخ اصلاح مقاله: ۱۳۹۶/۸/۸
تاریخ پذیرش مقاله: ۱۳۹۶/۱۲/۲۵
نام نویسنده مسئول: ۱۳۹۷/۳/۱۹
نشانی نویسنده مسئول: اسلام ناظمی

۱ -مقدمه

پیچیدگی و گستردگی سیستم‌های مدیریت پایگاه داده امروزی به حدی رسیده است که کنترل و نگهداری آن‌ها بسیار پرهزینه و زمان‌بر است و در بعضی از موارد از عهده نیروی انسانی خارج است. به همین علت در سال‌های اخیر مطالعات بسیاری در مورد اضافه نمودن قابلیت خودتطبیقی[۱] به این سیستم‌ها انجام شده است که حاصل آن طراحی و توسعه سیستم‌های مدیریت پایگاه داده خودمختار[۲] است. قابلیت مذکور، سیستم پایگاه داده را قادر به مدیریت و نگهداری از خود بدون دخالت انسان می‌سازد. سیستم‌های خودتطبیق(یا سیستم‌های خودمختار)، سیستم‌های خود مدیریتی[۳] هستند که از حلقه‌های کنترل بازخورد[۴] برای پایش، تحلیل، طرح‌ریزی و اجرا بر طبق تغییرات رخ داده در محیط خود، استفاده می‌کنند [۱]. بنابراین سیستم پایگاه داده خودمختار می‌تواند به تغییرات ایجاد شده در وضعیت و محیط عملیاتی خود پاسخ دهد و خود را با این تغییرات تطبیق نماید.

ویژگی خودتنظیمی[۵] یکی از ویژگی‌های پایگاه داده خودمختار است که اشاره به تنظیم خودکار شِمای پایگاه داده[۶] دارد. این ویژگی در پایگاه داده رابطه‌ای به خودکارسازی طراحی شِمای فیزیکی پایگاه داده می‌پردازد. مساله طراحی فیزیکی خودکار پایگاه داده[۷]، یافتن یک پیکربندی مناسب (مجموعه‌ای از ساختارهای فیزیکی) برای یک بارکاری[۸] مشخص(W) و بودجه ذخیره‌سازی مشخص(B) است که منجر به کمترین هزینه اجرای پرس‌وجوهای موجود در بارکاری می‌شود [۲]. مساله طراحی شِما برای پایگاه داده غیر رابطه‌ای از طراحی شِمای فیزیکی رابطه‌ای متفاوت است. طراحی شِما در پایگاه داده غیر رابطه‌ای فقط محدود به ساختارهای فیزیکی نمی‌شود و ازآنجایی‌که مدل داده منطقی هم بر مبنای پرس‌وجوهای برنامه قابل‌تغییر است نیز در برمی‌گیرد.

در پایگاه داده ستون‌گسترده[۹](ستون‌گرا) که نوع خاصی از پایگاه داده غیررابطه‌ای NoSQL است، جداولی از رکوردها ایجاد می‌شود که هر رکورد دارای یک کلید است و مجموعه ستون‌های هر رکورد، از پیش تعریف شده نیستند. همچنین هر رکورد می‌تواند دارای ستون‌های مختلفی نیز باشد. این جداول در پایگاه داده ستون‌گسترده، گروه ستون[۱۰] (CF) نامیده می‌شوند [۳]. بنابراین مساله اصلی طراحی شِما در این نوع از پایگاه داده NoSQL، تعیین گروهی از ستون‌ها و اطلاعاتی است که در هر گروه ستون برای یک بارکاری مشخص باید ذخیره شود تا هزینه اجرای پرس‌وجوهای بارکاری کمینه گردد [۴]. انتخاب یک شِمای خوب در این پایگاه داده وابسته به مدل داده مفهومی و بارکاری(پرس‌وجوهای) برنامه است [۵]. بنابراین با ایجاد تغییرات در بارکاری و مدل داده مفهومی برنامه، ممکن است شِمای طراحی شده دیگر از کارایی بالا و مناسب برخوردار نباشد و در نتیجه برای حفظ کارایی، ضرورت تطبیق با این تغییرات وجود دارد.

در زمینه خودکارسازی طراحی شِمای پایگاه داده غیررابطه‌ای تحقیقات کمی انجام شده است [۱۲-۴]. روش‌های اندک ارائه شده در این زمینه برون‌خط هستند و فقط به دنبال طراحی شِمای بهینه برای پایگاه داده غیررابطه‌ای می‌باشند. بنابراین قادر به تطبیق شِمای طراحی شده با تغییرات به وجود آمده در بارکاری و وضعیت سیستم نیستند. ازآنجاکه روش‌های برون‌خط نمی‌توانند اهداف سیستم پایگاه داده خودتنظیم را به‌درستی برآورده سازند، ارائه روش‌های برخط برای خودکارسازی طراحی شِمای پایگاه داده غیررابطه‌ای و تطبیق آن با تغییرات به‌عنوان یک چالش اصلی مطرح است. لازمه رسیدن به این هدف، پایش و تحلیل پیوسته بر بارکاری و وضعیت سیستم برای کشف تغییرات و فراخوانی بازتنظیمی شِما برای تطبیق با این تغییرات است که در این تحقیق به آن پرداخته می‌شود.

در ادامه، مقاله بدین‌صورت سازماندهی شده است: در بخش ۲ مهم‌ترین کارهای مرتبط با موضوع این پژوهش بررسی خواهند شد. در بخش ۳ چالش‌های موجود در تحقیقات پیشین و طرح مساله بیان می‌گردد. بخش ۴ شامل راهکار پیشنهادی برای کشف تغییرات بارکاری در پایگاه داده ستون‌گرا NoSQL است. در بخش ۵ نتایج حاصل از راهکار پیشنهادی تشریح می‌شود. در پایان نتیجه‌گیری مختصری در خصوص پژوهش و کارهای آینده در بخش ۶ معرفی شده‌اند.

۲ -مروری بر کارهای پیشین

در این بخش ابتدا مطالعات پیشین محققان در زمینه پایگاه داده‌های خودتنظیم که شامل راهکارهای ارائه شده برای طراحی خودکار شِمای پایگاه داده و تطبیق آن با تغییرات بارکاری است بررسی می‌گردد. سپس به بررسی مطالعات انجام شده در زمینه مدیریت بارکاری شامل ویژگی‌سازی، دسته‌بندی، مدل‌سازی و کشف تغییرات بارکاری پرداخته می‌شود.

۲ ۱ -مطالعات پیشین محققان در پایگاه داده خودتنظیم

در زمینه تنظیم خودکار طراحی فیزیکی برای پایگاه داده رابطه‌ای تحقیقات زیادی انجام شده است و محققان ابزارهای راهنمایی[۱۱] نظیر راهنمای طراحی IBM DB2 [۱۳, ۱۴]، راهنمای تنظیم SQL در اوراکل [۱۵] و راهنمای تنظیم پایگاه داده Microsoft SQL Server [۱۶] ارائه کرده‌اند که در محصولات تجاری پایگاه داده پیاده‌سازی شده‌اند. این راهنماها برون‌خط هستند و فقط به دنبال یافتن مجموعه بهینه‌ای از ساختارهای فیزیکی نظیر اندیس‌ها و دیدهای ذخیره شده برای بارکاری برنامه هستند. همچنین محققان به ارائه روش‌های برخط، نظیر [۲۱-۱۷]، در این زمینه پرداخته‌اند که قادر به نظارت پیوسته بر بارکاری و وضعیت سیستم و تنظیم پیکربندی ساختارهای فیزیکی پایگاه داده با تغییرات به وجود آمده در آن‌ها هستند. در [۲۲] مدلی بهبودیافته برای فرآیند طراحی شِمای منطقی پایگاه داده تحلیلی، ارائه می‌شود که ترکیبی از سه مدل دانه‌برفی، خوشه‌ستاره‌ای و ستاره‌ای به‌صورت سه‌لایه‌ای است. در [۲۳] با استفاده از الگوریتم فرهنگی ترکیبی، راهکاری برای انتخاب یک مجموعه دید مناسب از

بین همه دیدها، جهت ذخیره‌سازی دید در پایگاه داده تحلیلی ارائه شده است.

در سال ۲۰۱۶، پاولو و همکارانش در گروه تحقیقاتی پایگاه داده از دانشگاه کارنگی ملون، اولین سیستم مدیریت پایگاه داده خود-گرداننده[۱۲] بنام Peloton[۱۳] را توسعه داده‌اند. سیستم مدیریت پایگاه داده خود-گرداننده می‌تواند خودش را بدون هیچ‌گونه دخالت انسان پیکربندی، تنظیم و بهینه کند. Peloton یک سیستم مدیریت پایگاه داده رابطه‌ای است که برای عملیات خودمختار طراحی شده و قادر به کنترل تمام جنبه‌های سیستم پایگاه داده توسط یک مؤلفه طرح‌ریزی یکپارچه است [۲۴].

طراحی شِمای منطقی در پایگاه داده NoSQL فقط وابسته به شِمای مفهومی نیست، بلکه به پرس‌وجوهای برنامه هم وابسته است. بنابراین خودتنظیمی در پایگاه داده غیررابطه‌ای فقط محدود به ساختارهای فیزیکی نمی‌شود و ازآنجایی‌که مدل داده منطقی هم بر مبنای پرس‌وجوهای برنامه قابل‌تغییر هستند نیز در برمی‌گیرد. انتخاب یک شِمای مناسب برای پایگاه داده غیررابطه‌ای NoSQL، یک مساله مهم است که اصولاً به‌صورت دستی انجام می‌شود و در این زمینه فقط قوانین و خط‌مشی‌هایی از نمونه‌های عملی انجام شده، وجود دارد. ازاین‌رو در زمینه خودکارسازی طراحی شِمای پایگاه داده غیررابطه‌ای تحقیقات کمی انجام شده است [۱۲-۴]. روش‌های ارائه شده در این زمینه از فنون غیر نرمال‌سازی پایگاه داده برای جلوگیری از عملیات پیوند استفاده نموده‌اند و به دنبال طراحی شِمای بهینه‌ای از پایگاه داده برای بالا بردن کارایی پرس‌وجوها هستند. روش‌ها و ابزارهای اندک ارائه شده در این زمینه، برون‌خط هستند و قادر به تطبیق شِمای طراحی شده با تغییرات به وجود آمده در بارکاری و وضعیت سیستم نیستند. در ادامه به بررسی هر یک از این تحقیقات می‌پردازیم.

میئور و همکارانش در سال ۲۰۱۶ ابزاری بنام NoSE[۱۴] برای پیشنهاد طراحی شِمای بهینه پایگاه داده برای نوع خاصی از پایگاه داده NoSQL بنام ذخیره رکوردهای گسترش‌پذیر(ستون گسترده) معرفی نموده‌اند [۴]. ارائه کاملی از جزئیات عملکرد این ابزار در سال ۲۰۱۷ در [۹] بیان شده است. این روش مبتنی بر هزینه، از فرموله‌سازی مساله با برنامه‌ریزی عدد صحیح صفر و یک[۱۵] برای نگاشت از مدل داده مفهومی پایگاه داده به شِمای پایگاه داده استفاده می‌کند. مساله اصلی طراحی شِما در ذخیره رکوردهای گسترش‌پذیر، تعیین گروهی از ستون‌ها و اطلاعاتی است که در هر گروه ستون برای یک بارکاری مشخص باید ذخیره شود تا هزینه اجرای پرس‌وجوهای بارکاری کمینه گردد. راهنمای NoSE شِمای مفهومی داده‌های برنامه و بارکاری را به‌عنوان ورودی دریافت می‌کند. سپس با استفاده از فرموله‌سازی مساله با برنامه‌ریزی عدد صحیح صفر و یک و حل آن، مجموعه بهینه‌ای از گروه‌های ستون که هزینه اجرای پرس‌وجوها را کمینه می‌کنند و یک مجموعه از طرح‌های اجرایی (یک طرح برای هر پرس‌وجو) را به‌عنوان خروجی پیشنهاد می‌دهد. هر طرح شرح می‌دهد که چگونه برنامه باید از گروه‌های ستون در شِمای پیشنهادی برای پیاده‌سازی یک پرس‌وجو، استفاده کند.

واجک و همکارانش در [۱۰] قسمتی از شِمای رابطه‌ای پایگاه داده تویتر را استفاده کرده‌اند و با استفاده از غیر نرمال‌سازی آن به‌طور خودکار به شناسایی شِمای ستون‌گرا NoSQL با کمترین هزینه پرداخته‌اند. این روش همانند روش [۴] با یک شِمای مفهومی شروع و بهینه‌سازی شِمای NoSQL را ارائه می‌دهد. همچنین واجک و همکارانش در تحقیقی مشابه با کار قبلی، الگوریتمی برای یافتن خودکار شِمای بهینه در پایگاه داده ستون‌گرا NoSQL ارائه نموده‌اند [۱۱]. این الگوریتم با دریافت مجموعه‌ای از پرس‌وجوهای تعریف شده و یک شِمای اولیه رابطه‌ای و با استفاده از روش غیر نرمال‌سازی شِما، به یافتن شِمای بهینه می‌پردازد. در این روش پرس‌وجوها با زبان قیود شی بیان می‌شوند که به‌راحتی قابل تبدیل به مدل مفهومی از رابطه‌ها هستند. در این الگوریتم ابتدا شِمای رابطه‌ای اولیه به‌طور خودکار غیر نرمال‌سازی می‌شود و تمام شِماهای ممکن (رابطه‌های غیرنرمال) ایجاد می‌شوند. سپس بر مبنای پرس‌وجوهای تعریف شده، اندیس‌های تک-ستونه برای جداول انتخاب می‌شوند. سرانجام با استفاده از یک تابع هزینه، هزینه اجرای پرس‌وجوها برای هر شِما محاسبه شده و شِمایی با کمترین هزینه به‌عنوان شِمای بهینه انتخاب می‌شود.

بِرمبَچ و همکارانش در [۵]، یک روش سیستماتیک برای طراحی خودکار شِمای پایگاه داده ستون‌گرا NoSQL ارائه نموده‌اند. هدف اصلی این روش بهینه‌سازی شِمای پایگاه داده برای پرس‌وجوهای خواندنی است. بدین‌صورت که هر پرس‌وجوی خواندن تنها با یک درخواست به پایگاه داده اجرا شود. این روش شامل دو مرحله ایجاد شِمای خودکار از مدل مفهومی اولیه و ارزیابی شِما است. در مرحله ایجاد شِما، ابتدا با تحلیل پرس‌وجوهای انتخاب و پیوند و روش‌های غیر نرمال‌سازی تمام شِماهای ممکن شناسایی می‌شود. این شِماها شامل جداول ستون‌گرایی هستند که منجر به اجرای پرس‌وجوها، تنها با یک درخواست به پایگاه داده می‌شوند. سپس با حذف یا ادغام جداول در شِماها بر طبق شرط‌های تعریف شده، این شِماها بهینه‌سازی می‌شوند. در مرحله ارزیابی شِما با استفاده از مجموعه‌ای از متریک‌ها و یک تابع امتیازدهی، بهترین شِما از بین شِماهای پیشنهادی انتخاب می‌شود.

همان‌طورکه گفته شد طراحی شِمای پایگاه داده NoSQL مبتنی بر پرس‌وجوها است، بنابراین پرس‌وجوها نقش مهمی را ایفا می‌کنند. تحقیقاتی نظیر [۸-۶, ۱۲] در زمینه طراحی شِمای منطقی مبتنی بر بارکاری در پایگاه داده NoSQL و پیاده‌سازی فیزیکی آن، ارائه شده است. نتایج آزمایشات نشان می‌دهد که این روش‌ها، کارایی پرس‌وجوها (زمان پردازش پرس‌وجوها) را توسط کاهش تعداد دسترسی به پایگاه داده NoSQL بهبود می‌بخشند.

روش‌های ارائه شده در [۸, ۷] شامل فرایند تبدیل شِمای مفهومی به شِمای منطقی در پایگاه داده NoSQL هستند که در آن‌ها از

اطلاعات بارکاری برای تعیین شِمای منطقی بهینه استفاده شده است. روش ارائه شده در [۸] شامل فرایند تبدیل شِمای مفهومی به شِمای منطقی در
پایگاه داده سندگرا NoSQL و پیاده‌سازی فیزیکی آن است. این روش برای مدل‌سازی و تحلیل بارکاری بر مبنای روش ارائه شده در [۲۵] است. بر طبق این روش، برای جمع‌آوری و تحلیل اطلاعات بارکاری، طبق قانون ۸۰-۲۰، بر روی ۲۰ درصد از عملیات پرتکرار بارکاری تمرکز می‌شود. در این روش تحلیل بارکاری شامل شناسایی موجودیت‌ها و ارتباط‌هایی در مدل مفهومی است که توسط پرس‌وجوهای بارکاری مکرراً استفاده شده‌اند. همان‌طورکه گفته شد، این روش توسط کاهش تعداد دسترسی به پایگاه داده، کارایی پرس‌وجوها در روی مستندات NoSQL را بهبود می‌بخشد.

روش ارائه شده در [۶] به طراحی شِمای منطقی برای پایگاه داده تحلیلی در پایگاه داده ستون‌گرا NoSQL پرداخته است. در حقیقت این روش، پایگاه داده تحلیلی رابطه‌ای را به پایگاه داده تحلیلی ستون‌گرا NoSQL تبدیل می‌کند. این روش از تکنیک خوشه‌بندی K-means برای طراحی بهتر شِمای منطقی پایگاه داده ستون‌گرا(مجموعه‌ای از گروه‌های ستون) از جداول پایگاه داده تحلیلی، استفاده کرده است.

یانگ و همکارانش در [۱۲] برای یافتن شمای بهینه پایگاه داده ستون‌گرای Hbase برای پرس‌وجوهای OLAP، یک الگوریتم تکاملی جدید طراحی نموده‌اند. این روش بر اساس پرس‌وجوهای برنامه و با استفاده از الگوریتم تکاملی طراحی شده، ستون‌ها را گروه‌بندی می‌کند و شِمای بهینه‌ای از گروه‌های ستون را پیدا می‌کند. با استفاده از این روش کارایی خواندن پرس‌وجوهای OLAP (متوسط زمان پاسخ) به‌طور چشمگیری بهبود یافته است.

۲ ۲ -مطالعات پیشین محققان در مدیریت بارکاری

پایگاه داده خودمختار با فراهم آوردن عملکرد خود مدیریتی، منجر به کاهش هزینه کل مالکیت برای سیستم پایگاه داده می‌شود. تصمیمات خود مدیریتی نظیر تنظیم خودکار شِمای پایگاه داده وابسته به بارکاری پایگاه داده است. یکی از مسائل مهم در تحقق تنظیم خودکار پایگاه داده، پایش و تحلیل بارکاری برای کشف تغییرات و تطبیق طراحی (بازتنظیمی) شِمای پایگاه داده با این تغییرات است. در حقیقت بازتنظیمی شِمای پایگاه داده نیازمند شناسایی تغییرات در بارکاری است. بنابراین پایش و تحلیل بارکاری برای شناخت بهتر بارکاری یک نیاز مبرم است. مطالعات مرتبط در این زمینه شامل ویژگی‌سازی، دسته‌بندی، مدل‌سازی و کشف تغییرات بارکاری است. در ادامه به مروری بر کارهای انجام شده در این زمینه پرداخته می‌شود.

یک تحقیق اولیه که به شناخت و درک عمیق از بارکاری پایگاه داده رابطه‌ای پرداخته است،REDWAR[۱۶] [۲۶] است. REDWAR یک تحلیلگر برای ویژگی‌سازی بارکاری در محیط رابطه‌ای DB2 است که به مطالعه و تحلیل ساختار جملات SQL و رفتار زمان اجرای پرس‌وجوها می‌پردازد. REDWAR شامل دو مؤلفه تحلیلگر SQL و تحلیلگر ردیابی است. تحلیلگر SQL برای تحلیل جملات SQL و اطلاعات کاتالوگ است و تحلیلگر ردیابی برای پردازش ردیابی و جمع‌آوری آمارهای زمان اجرا است. نتایج تحلیل REDWAR می‌تواند برای طراحی فیزیکی پایگاه داده و یا برای ایجاد بارکاری معیار سنجش[۱۷] به‌منظور ارزیابی طراحی بکار رود.

ازآنجایی‌که شناسایی نوع بارکاری نقش مهمی در تنظیم طراحی فیزیکی پایگاه داده رابطه‌ای دارد، اِلنافر و همکارانش در [۲۷] روشی برای شناسایی خودکار نوع بارکاری پردازش تراکنشی برخط (OLTP) از سیستم پشتیبانی تصمیم[۱۸] (DSS) ارائه داده‌اند. در این روش یک مدل دسته‌بندی بر مبنای ویژگی‌هایی از بارکاری که جداکننده نوع OLTP از DSS هستند، ایجاد می‌شود. سپس از مدل ایجاد شده برای شناسایی تغییر در نوع بارکاری از OLTP به DSS و برعکس استفاده می‌شود. مدل دسته‌بندی بر مبنای درخت تصمیم است که در آن DSS و OLTP برچسب دسته‌ها هستند و ویژگی‌های بارکاری در گره‌های درخت نشان داده می‌شوند. این روش فقط محدود به کشف تغییرات در دامنه نوع بارکاری OLTP از DSS است و ترکیب‌های مختلف بارکاری را در یکی از این دو دسته قرار می‌دهد. فعال نگه داشتن دسته‌بند بارکاری و پایش پیوسته سیستم برای کشف تغییرات در نوع بارکاری، سربار زیادی را به سیستم وارد می‌نماید و منجر به کاهش کارایی سیستم پایگاه داده می‌شود. بنابراین النافر و همکارانش در تحقیق دیگری [۲۸] چارچوب PSP[۱۹] را ارائه داده‌اند که با مدل دسته‌بندی بارکاری یکپارچه شده تا توسط آن سیستم پایگاه داده خودمختار بتواند پیکربندی خود را به‌طور مؤثرتری تنظیم نماید. در این روش بجای پایش پیوسته بارکاری، فقط در طول بازه‌های زمانی خاصی که توسط چارچوب PSP پیشنهاد می‌شود، بارکاری پایش می‌شود.در حقیقت این چارچوب زمانی که یک تغییر در نوع بارکاری اتفاق خواهد افتاد را پیش‌بینی کرده و پایش برخط را به این بازه‌های زمانی خاص محدود می‌کند.

مدل ارائه شده توسط زِودیو و همکارانش در [۲۹] به ویژگی‌سازی بارکاری و شناسایی متغیرهای وضعیت پایگاه داده رابطه‌ای که تحت تأثیر تغییرات بارکاری هستند، می‌پردازد. این مدل شامل دو مؤلفه کشف بارکاری و دسته‌بندی بارکاری است. مؤلفه کشف، بارکاری ورودی را تحلیل می‌کند و متغیرهای وضعیت پایگاه داده که توسط مؤلفه دسته‌بندی برای دسته‌بندی بارکاری به OLTP و DSS استفاده می‌شوند را ذخیره می‌کند. سپس دسته‌بندی بارکاری از طریق الگوریتم خوشه‌بندی سلسله مراتبی و الگوریتم طبقه‌بندی و درخت رگرسیون[۲۰] انجام می‌شود. سرانجام برای کارایی بهتر سیستم پایگاه داده، پارامترهای پیکربندی سیستم می‌توانند بر طبق نوع بارکاری دسته‌بندی شده، تنظیم شوند که در این تحقیق به آن پرداخته نمی‌شود.

هولز و همکارش در [۳۰] به ارائه روشی برای پایش و تحلیل پیوسته و سبک-وزن بارکاری در پایگاه داده رابطه‌ای خودمختار پرداخته‌اند. این روش که مبتنی بر مدل‌های n-gram است به کشف تغییرات بارکاری می‌پردازد. در این روش با تحلیل سبک-وزن بارکاری، سربار تحلیل برای بارکاری ثابت و بدون تغییر را تقریباً از بین می‌برد، بنابراین فقط با کشف تغییرات مهم در بارکاری، تحلیل بازپیکربندی سنگین‌وزن پایگاه داده انجام می‌شود. سناریوهای تغییر بارکاری که در این تحقیق موردنظر هستند، عبارت‌اند از: انتشار نسخه‌های جدید از برنامه‌ها، استقرار برنامه‌های جدید، تغییرات دوره‌ای در بارکاری، تغییرات ناشی از استفاده برنامه.

در [۳۱] به شناسایی و خوشه‌بندی جلسات کاربر از بارکاری پایگاه داده رابطه‌ای پرداخته شده است. در این روش از مدل‌های n-gram برای ایجاد مدل آماری از بارکاری، استفاده شده است. این مدل برای شناسایی جلسات کاربر از ردیابی جملات SQL بکار می‌رود. سپس یک الگوریتم خوشه‌بندی مبتنی بر فاصله برای گروه‌بندی جلسات کاربر در کلاس‌های جلسه متفاوت، ارائه شده است. نتایج این روش می‌تواند برای پیش‌بینی پرس‌وجوهای ورودی بر مبنای پرس‌وجوهایی که قبلاً تثبیت شده‌اند، بکار رود؛ این پیش‌بینی می‌تواند برای بهبود کارایی پایگاه داده توسط بازنویسی پرس‌وجو، استفاده شود.

در [۳۲] روشی برای مدیریت بارکاری بر مبنای نوع بارکاری OLTP و DSS در سیستم پایگاه داده رابطه‌ای ارائه شده است. هدف اصلی این تحقیق طراحی و توسعه یک چارچوب مدیریت بارکاری خودمختار است که می‌تواند بدون دخالت انسان بارکاری پایگاه داده را مدیریت کند. این روش شامل سه مؤلفه ویژگی‌سازی بارکاری، زمان‌بندی بارکاری و کشف زمان‌های بیکاری CPU برای مدیریت فعالانه بارکاری است. مؤلفه ویژگی‌سازی بارکاری با استفاده از روش استدلال مبتنی بر نمونه[۲۱] و منطق فازی، بارکاری را بدون تأثیر بر زمان اجرا، ویژگی‌سازی و دسته‌بندی می‌نماید. این مؤلفه با شناسایی پارامترهای مؤثر در شناسایی نوع بارکاری، بارکاری ورودی را به OLTP و DSS طبقه‌بندی می‌کند. در روش استدلال مبتنی بر نمونه برای شناسایی نوع بارکاری ورودی از مقادیر نمونه‌های ذخیره شده قبلی برای پارامترها استفاده می‌شود [۳۳]. در این روش، در بازنمایی و بازیابی نمونه‌ها (مقیاس شباهت) از منطق فازی استفاده شده است تا در مدیریت ابهام و عدم قطعیتی که در نمونه‌های جمع‌آوری شده وجود دارد به سیستم کمک کرده و کارایی آن را افزایش دهد [۳۳]. در کارهای قبلی برای شناسایی نوع بارکاری، ویژگی‌سازی و دسته‌بندی بارکاری با اجرای بارکاری انجام می‌شود ولی در این روش قبل از شروع اجرای بارکاری این کار انجام می‌شود. نتایج آزمایشات از روش ارائه شده در این تحقیق برای زمان‌بندی بارکاری نشان می‌دهد که متوسط زمان انتظار بارکاری در این روش بهتر از روش‌های زمان‌بندی بارکاری مشهو نظیر FIFO و SJR است.

سازماندهی داده‌ها در پایگاه داده NoSQL نیازمند تصمیمات طراحی مهمی است، زیرا این تصمیمات طراحی تأثیر بسزایی در نیازمندی‌های کیفیت نظیر مقیاس‌پذیری و کارایی دارد [۳۴]. طراحی شِمای پایگاه داده NoSQL مبتنی بر پرس‌وجوها است، بنابراین پرس‌وجوها نقش مهمی را ایفا می‌کنند. در تحقیقاتی نظیر [۸-۶, ۱۲] برای طراحی شِمای منطقی در پایگاه داده NoSQL، به مدل‌سازی و تحلیل پرس‌وجوهای بارکاری پرداخته شده است. این روش‌ها با استفاده از اطلاعات بارکاری توانسته‌اند شِمای منطقی بهینه‌ای از پایگاه داده NoSQL را ارائه دهند که باعث بهبود کارایی پرس‌وجوها(زمان پردازش پرس‌وجوها) می‌شود. توضیح کاملی از مدل‌سازی و تحلیل بارکاری در این روش‌ها در بخش ۲-۱ بیان شد.

۳ -چالش‌های موجود و طرح مساله

ازآنجاکه تمرکز این تحقیق بر ویژگی خودتنظیمی شِمای پایگاه داده غیررابطه‌ای است، ابتدا در این بخش به بررسی چالش‌های موجود در تحقیقات پیشین پرداخته می‌شود تا چرایی و لزوم انجام پژوهش مشخص گردد. سپس به بیان طرح مساله پرداخته خواهد شد.

راه‌حل‌های ارائه‌شده در زمینه پایگاه داده خودتنظیم غیررابطه‌ای دارای نقص‌ها و کاستی‌هایی هستند و هنوز اهداف محاسبات خودکار را به‌طور کامل برآورده نمی‌سازند. برخی از چالش‌هایی که در تحقیقات پیشین به چشم می‌خوردند به شرح زیر هستند:

- **بهینه‌سازی شِمای پایگاه داده**: هدف اصلی در طراحی شِمای پایگاه داده غیررابطه‌ای، ارائه شِمای بهینه برای افزایش کارایی پرس‌وجوهای برنامه است. به علت محدود بودن تحقیقات در این زمینه، هنوز ارائه ابزارهایی برای بهینه‌سازی شِما غیررابطه‌ای به‌عنوان یک چالش مهم مطرح است.
- **ارائه روش‌های برخط برای تطبیق خودکار شِمای طراحی شده با تغییرات:** همان‌طورکه گفته شد انتخاب شِمای بهینه در پایگاه داده غیررابطه‌ای وابسته به مدل مفهومی و بارکاری برنامه است. بنابراین با ایجاد تغییرات در بارکاری و مدل داده مفهومی برنامه، ضرورت تطبیق با این تغییرات وجود دارد. روش‌های ارائه شده در زمینه طراحی شِما پایگاه داده غیررابطه‌ای برون‌خط هستند و قادر به تطبیق شِمای طراحی شده با تغییرات به وجود آمده در مدل مفهومی و بارکاری برنامه نیستند. در حال حاضر هیچ روش برخطی برای تطبیق خودکار شِمای پایگاه داده غیررابطه‌ای NoSQL با تغییرات ارائه نشده است که با توجه به نقاط قوت این روش نسبت به روش برون‌خط، تأمین آن به‌عنوان یک چالش مطرح است.
- **طراحی الگوریتم‌های تحلیل بازپیکربندی سبک‌وزن برای روش‌های برخط:** در روش‌های برخط باید به‌طور پیوسته

کنترل شود که آیا پیکربندی فعلی برای بارکاری جاری مناسب است یا خیر. استفاده از تحلیل بازپیکربندی سنگین‌وزن منجر به سربار غیرقابل‌قبولی می‌شود. درحالی‌که تحلیل سبک‌وزن بارکاری، امکان تطبیق سریع پیکربندی با تغییرات بارکاری را فراهم کرده و سربار تحلیل برای بارکاری ثابت و بدون تغییر را تقریباً از بین می‌برد. بنابراین تحلیل بازپیکربندی سبک‌وزن در روش‌های برخط برای کاهش سربار امری مهم است.

- **سربار:** روش‌های برخط، راه‌حل‌های خودکار کامل را ارائه می‌دهند که اهداف محاسبات خودمختار را برآورده می‌سازد. این روش‌ها از یک‌سو سربار نگهداری سیستم را برای مدیر پایگاه‌داده کاهش می‌دهند، ولی از سوی دیگر باعث تحمیل سربار محاسباتی اضافی به سیستم هستند. بنابراین چالش اصلی در آن‌ها، ارائه راهکارهایی نظیر طراحی روش‌های تحلیل سبک وزن برای کاستن این سربار تا حد ممکن است.
- **افزونی واکنش:** روش‌های برخط باید در برابر تغییرات و نوسانات کم در بارکاری و وضعیت سیستم مقاوم باشند و از واکنش‌های اضافی در برابر این تغییرات جلوگیری کنند. بنابراین چالشی دیگر در خودتنظیمی پایگاه داده، توسعه روش‌هایی است که قادر به پیش‌بینی اثرات تغییر پیکربندی و جلوگیری از افزونی واکنش هستند.

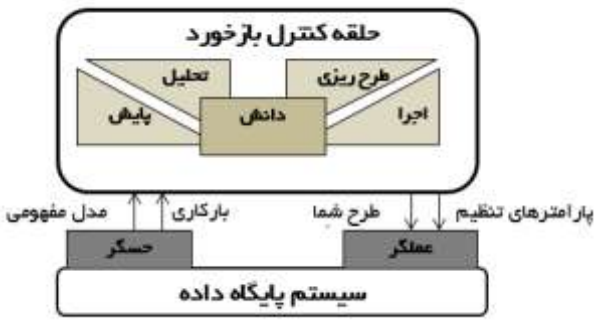
همان‌طورکه مشاهده می‌شود یکی از چالش‌های اصلی در زمینه خودتنظیمی پایگاه داده غیررابطه‌ای، ارائه روش‌های برخط برای تطبیق خودکار شِمای طراحی شده با تغییرات بارکاری و مدل مفهومی برنامه است. لازمه رسیدن به این هدف و تنظیم تمام‌خودکار پایگاه داده، پایش و تحلیل پیوسته بر بارکاری و مدل مفهومی برای کشف تغییرات در آن‌ها و تطبیق طراحی شِمای پایگاه داده با این تغییرات است. در این پژوهش کشف تغییرات فقط محدود به تغییرات بارکاری است و کشف تغییرات در مدل مفهومی برنامه به کارهای آتی سپرده می‌شود. ازاین‌رو در این تحقیق یک حلقه کنترل بازخورد مطابق با چرخه کنترل خودمختار IBM [۳۵]، برای پایش پیوسته و تحلیل سبک وزن بارکاری در پایگاه داده ستون‌گرا NoSQL ارائه می‌شود. این حلقه توصیف کننده یک الگوی طراحی برای کشف تغییرات بارکاری بکار می‌رود که لازمه بازتنظیمی خودکار شِمای پایگاه داده است.

همان‌طورکه در شکل ۱ مشاهده می‌شود، حلقه کنترل بازخورد از یک کنترل‌کننده که شامل فرایند خودتنظیمی می‌باشد و یک عنصر مدیریت شده (سیستم پایگاه داده) تشکیل شده است. ازآنجایی‌که انتخاب یک شِمای خوب در پایگاه داده ستون‌گرا NoSQL وابسته به مدل داده مفهومی و بارکاری (پرس‌وجوهای) برنامه است، عنصر کنترل‌کننده از طریق حسگرها به‌طور پیوسته به پایش و جمع‌آوری اطلاعات بارکاری و مدل مفهومی پایگاه داده و تحلیل آن می‌پردازد. هنگامی‌که این اطلاعات نیاز به تنظیم مجدد شِمای پایگاه داده را نشان دهند، کنترل‌کننده برای انتخاب شِمای مناسب، طرح‌ریزی می‌کند و سپس از طریق عملگرها، شِمای پیشنهادی و پارامترهای تنظیم شِما را به سیستم پایگاه داده اعمال می‌کند. تمرکز اصلی این تحقیق بر روی مراحل پایش و تحلیل برای کشف تغییرات در بارکاری است. کشف تغییرات در مدل مفهومی پایگاه داده و همچنین ارائه ابزاری برای انتخاب خودکار شِمای مناسب از پایگاه داده در مرحله طرح‌ریزی به کارهای آتی سپرده می‌شود.

۴ -راهکار پیشنهادی

همان‌طورکه گفته شد، در این پژوهش برای کشف تغییرات بارکاری در پایگاه داده ستون‌گرا NoSQL یک حلقه کنترل بازخورد ارائه شده است. در این بخش به توصیف معماری حلقه کنترل بازخورد پیشنهادی و توصیف مولفه‌ها و ارتباط آن‌ها پرداخته می‌شود.

شکل ۲ دید مفهومی از معماری ارائه شده برای حلقه کنترل بازخورد پیشنهادی را نشان می‌دهد. مطابق این شکل، مرحله پایش شامل پایش بارکاری[۲۲] است که به‌طور پیوسته به مشاهده جملات پرس‌وجوی (رویدادهای) بارکاری و استخراج ویژگی‌های مرتبط با آن‌ها می‌پردازد. سپس بردارهای ویژگی حاصل شده به مرحله تحلیل ارسال می‌شوند. در مرحله تحلیل به خوشه‌بندی رویدادهای بارکاری و کشف تغییرات حاصل شده در آن در طی زمان پرداخته شده است. با کشف تغییرات از مرحله تحلیل، منطق خودتنظیمی برای طراحی مجدد شِما و انتخاب شِمایی متناسب با تغییرات بارکاری از مرحله طرح‌ریزی رهانا می‌شود. همان‌طورکه گفته شد، تمرکز اصلی این تحقیق بر روی مراحل پایش و تحلیل برای کشف تغییرات در بارکاری است. در ادامه به توصیف مولفه‌ها و ارتباط آن‌ها در معماری ارائه شده برای این مراحل می‌پردازیم.



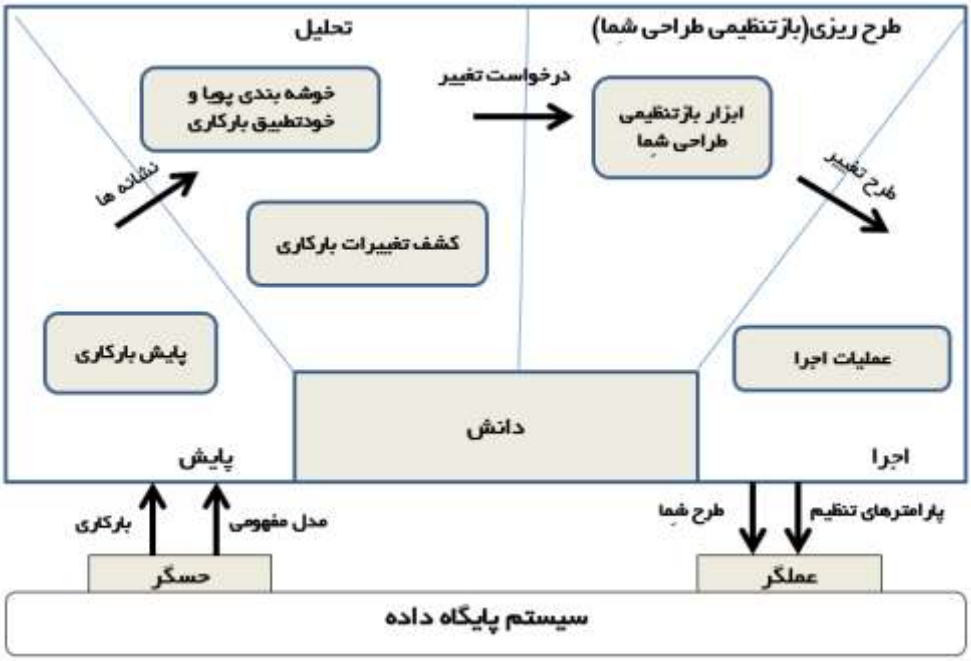
شکل ۱: الگوی طراحی حلقه کنترل بازخورد

۴ ۱ -معماری پیشنهادی برای مرحله پایش

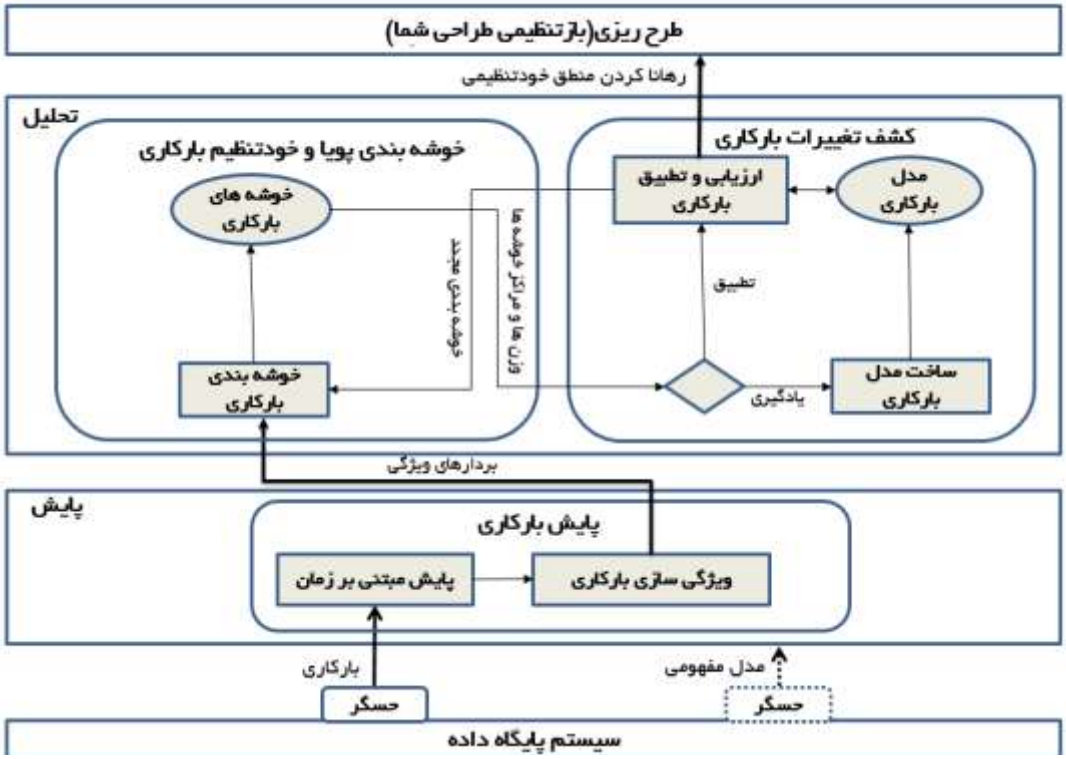
مرحله پایش بارکاری از طریق حسگرها به‌طور پیوسته به پایش و جمع‌آوری اطلاعات بارکاری پایگاه داده و تعیین ویژگی‌های آن می‌پردازد. پایش اطلاعات به دو نوع رویکرد مختلف پایش مبتنی بر رویداد[۲۳] و پایش مبتنی بر زمان[۲۴] تقسیم می‌شود [۳۶]. پایش مبتنی

بر رویداد توسط عامل خارجی رهانا می‌شود، درحالی‌که پایش مبتنی بر زمان وابسته به یک عامل داخلی است و در بازه‌های زمانی منظم پایش اطلاعات انجام می‌شود [۳۶].

مطابق شکل ۳، مؤلفه «پایش مبتنی بر زمان» در بازه‌های زمانی منظم، به‌طور پیوسته به مشاهده جملات پرس‌وجوی بارکاری می‌پردازد. ایجاد تغییرات در بارکاری بر پارامترهای پیکربندی مختلفی از سیستم پایگاه داده تاثیرگذار است و در نتیجه راه‌حل‌های ارائه شده در این زمینه بر یک کار مدیریتی خاص مانند تطبیق شِمای پایگاه داده، مدیریت حافظه و جمع‌آوری آمارها متمرکز هستند. برای پایش و جمع‌آوری اطلاعات بارکاری، منابع اطلاعاتی مختلفی از بارکاری موجود است. در



شکل ۲: دید مفهومی از حلقه کنترل بازخورد پیشنهادی



شکل ۳: معماری پیشنهادی برای حلقه کنترل بازخورد

مدیریت حافظه خودکار این منبع وابسته به نرخ‌های برخورد حافظه و در مکانیزم‌های جمع‌آوری آمارهای خودکار وابسته به متریک‌های بهینه‌ساز است. در مساله انتخاب شِمای پایگاه داده، منبع اطلاعات بارکاری شامل مجموعه‌ای از جملات پرس‌وجو است. با توجه به اینکه فقط نیازمند کشف تغییراتی در بارکاری هستیم که منجر به تغییر طراحی شِمای پایگاه داده
می‌شوند، پایش در سطح جملات پرس‌وجو مناسب‌تر است. در حقیقت برای کاهش سربار حاصل از پایش پیوسته تا حد امکان، پایش به دامنه خاصی؛ پایش در سطح جملات پرس‌وجو که مناسب برای کشف تغییرات بارکاری است؛ محدود شده است.

مؤلفه «ویژگی‌سازی بارکاری[۲۵]» به استخراج ویژگی‌های مرتبط با رویدادهای بارکاری (جملات پرس‌وجو) می‌پردازد. ارائه تمام جملات پرس‌وجوی مشاهده شده در بارکاری با متن جملات متمایز به مدل بارکاری باعث ایجاد مدلی با اندازه غیرقابل‌قبول در مرحله تحلیل می‌شود. اندازه بسیار بزرگ مدل، از یک‌سو مانع تحلیل سبک وزن بارکاری است و از سوی دیگر کشف تغییرات را به نوسانات و تغییرات جزئی در بارکاری حساس می‌کند. بنابراین رویدادهای بارکاری باید بر طبق ویژگی‌های مشخص، گروه‌بندی شوند. انتخاب ویژگی‌ها وابسته به هدف تحلیل بارکاری است. ازآنجایی‌که هدف تحلیل در این تحقیق، کشف تغییرات بارکاری در سطح جملات پرس‌وجو است، ویژگی‌های مرتبط با متن جملات پرس‌وجو موردنظر است. جدول ۱ ویژگی‌های انتخاب شده برای کشف تغییرات بارکاری را نشان می‌دهند. با توجه به اینکه هدف از کشف تغییرات بارکاری بازتنظیمی شِما برای پیشنهاد مجموعه مناسبی از گروه‌های ستون جهت تطبیق با این تغییرات است، پرس‌وجوها بر مبنای گروه‌های ستون نمی‌توانند بیان شوند زیرا در ابتدا مشخص نمی‌باشند. بنابراین پرس‌وجوهای بارکاری تحت مدل مفهومی بیان می‌شوند و در نتیجه ویژگی‌ها بر مبنای مدل مفهومی تعیین شده‌اند. خروجی مؤلفه «ویژگی‌سازی بارکاری» بردارهای ویژگی حاصل شده از رویدادهای بارکاری است که به مرحله تحلیل برای گروه‌بندی ارسال می‌شوند.

جدول ۱: ویژگی‌های انتخاب شده برای کشف تغییرات بارکاری

ویژگی	نوع ویژگی	دامنه ویژگی
نوع عملیات	کیفی-اسمی	مقدار رشته‌ای ساده
نام موجودیت‌ها	کیفی-اسمی	مجموعه‌ای از مقادیر رشته‌ای
صفت‌های تصویر	کیفی-اسمی	مجموعه‌ای از مقادیر رشته‌ای
صفت‌های انتخاب	کیفی-اسمی	مجموعه‌ای از مقادیر رشته‌ای

۴ ۴ -معماری پیشنهادی برای مرحله تحلیل

مطابق شکل ۳ مرحله تحلیل شامل دو بخش «خوشه‌بندی پویا و خودتطبیق بارکاری[۲۶]» و «کشف تغییرات بارکاری[۲۷]» است. در ادامه به توضیح هر یک از این بخش‌ها پرداخته می‌شود.

در بخش «خوشه‌بندی پویا و خودتطبیق بارکاری» با استفاده از تکنیک‌های داده‌کاوی و یادگیری ماشین، به گروه‌بندی رویدادهای بارکاری بر اساس ویژگی‌های تعیین شده در مرحله قبل، می‌پردازیم. گروه‌بندی رویدادهای بارکاری با قرار دادن رویدادهای مشابه در یک گروه، منجر به کاهش تنوع رویدادها و جلوگیری از ایجاد مدل بارکاری با اندازه بسیار بزرگ و تضمین تحلیل سبک وزن بارکاری در کشف تغییرات می‌شود. دسته‌بندی و خوشه‌بندی ازجمله رایج‌ترین روش‌های داده‌کاوی برای گروه‌بندی داده‌ها هستند. ازآنجایی‌که روش‌های دسته‌بندی نظیر درخت تصمیم و شبکه عصبی نیازمند مجموعه داده آموزشی برچسب‌دار و کلاس‌های مشخص و تعریف شده هستند، باعث تحمیل تلاش اضافی بر دوش مدیر پایگاه داده و برآورده نساختن اهداف خودمختاری می‌شوند. بنابراین برای گروه‌بندی بارکاری، مناسب نیستند.

در روش خوشه‌بندی، داده‌ها بر اساس یک معیار فاصله(شباهت) به مجموعه‌ای از خوشه‌ها گروه‌بندی می‌شوند. در این روش تعریف خوشه‌ها به‌صورت خودکار و فقط وابسته به داده‌های مشاهده شده است و درواقع یک روش آموزش با مشاهده‌هاست و نه آموزش با مثال‌ها [۳۷]. بنابراین در این تحقیق برای گروه‌بندی رویدادهای بارکاری از روش‌های خوشه‌بندی استفاده می‌نماییم. هدف از خوشه‌بندی بارکاری، قرار دادن هر نوع جمله (رویداد) در یک خوشه است. به‌طوری‌که هر خوشه شامل مجموعه جملات پرس‌وجویی است که به موجودیت‌های یکسانی دسترسی دارند و دارای نوع عملیات، صفات تصویر و صفات انتخاب یکسانی هستند. همچنین به هر خوشه وزنی نسبت داده می‌شود که برابر با تعداد پرس‌وجوهای تخصیص‌یافته به آن خوشه است. در حقیقت این وزن برابر با فراوانی نسبی (تعداد تکرار) آن جمله پرس‌وجو در بارکاری است. بنابراین با این روش بر اساس ویژگی‌های مشخص‌شده در مرحله قبل، رویدادهای بارکاری (جملات پرس‌وجو) خوشه‌بندی می‌شوند و جملات مرتبط به هم در یک دسته قرار می‌گیرند.

برای خوشه‌بندی بارکاری باید اهداف و نیازمندی‌هایی که در ادامه ذکر شده‌اند، مورد ملاحظه قرار گیرد. اولاً به علت پایش پیوسته بارکاری، با مجموعه‌ای از رویدادهای بارکاری ایستا روبرو نیستیم. بنابراین یک بارکاری، جریانی از رویدادهای ورودی است که باید به‌طور خودکار به خوشه‌ها تخصیص داده شوند. در نتیجه نیازمند روش خوشه‌بندی پویا بارکاری هستیم. دوما بارکاری ورودی در طی زمان دچار تغییرات می‌شود، ازاین‌رو روش خوشه‌بندی باید قابلیت به‌روزرسانی و آموزش مجدد را برای واکنش نشان دادن به تغییرات را داشته باشد. سوما تحلیل بارکاری برای کشف تغییرات، هیچ دانشی از

جملات اصلی پرس‌وجو که به خوشه‌ها نگاشت شده‌اند، ندارد و تحلیل فقط بر مبنای اطلاعات خوشه‌ها است. به همین علت خوشه‌بندی بارکاری باید سازگاری نتایج را تضمین نماید تا از تحلیل نادرست بارکاری جلوگیری شود. این بدین معناست که جملاتی از یک نوع یکسان باید در طول عمر فرایند خوشه‌بندی به خوشه یکسانی نگاشت شوند.

الگوریتم‌های خوشه‌بندی افرازی برای رویدادهای بارکاری مناسب‌تر هستند. زیرا آن‌ها بهترین عملکرد را برای شناسایی خوشه‌های مستقل و به‌خوبی جداشده[۲۸] از خود نشان می‌دهند و منجر به ایجاد خوشه‌های همگرا (همسانگرد) می‌شوند. الگوریتم خوشه‌بندی افرازی K-means، دارای پیچیدگی زمانی خوبی است ((O(m)) و برای مجموعه داده‌های بسیار بزرگ مناسب است [۳۸]. ازآنجایی‌که با جریانی از رویدادهای بارکاری روبرو هستیم که همواره در حال تغییر هستند، باید از الگوریتم‌های خوشه‌بندی برخط استفاده کرد. اکثر الگوریتم‌های K-means برخط نظیر [۳۹،۴۰] که برای خوشه‌بندی جریانی از داده‌ها ارائه شده‌اند، تمام اهداف ذکر شده برای خوشه‌بندی رویدادهای بارکاری را برآورده نمی‌سازند. این الگوریتم‌ها به‌طور پیوسته مراکز خوشه‌ها را با رویدادهای مشاهده شده، تطبیق می‌دهند و بنابراین نیازمندی سازگاری را برآورده نمی‌سازند. بنابراین در ادامه یک فرایند خوشه‌بندی برخط ارائه می‌شود که اهداف و نیازمندی‌های ذکر شده برای خوشه‌بندی جریانی از رویدادهای بارکاری را برآورده می‌سازد.

فرایند خوشه‌بندی ارائه شده، شامل سه حالت یادگیری، ثابت و تطبیقی است. ابتدا فرایند خوشه‌بندی در حالت اولیه یادگیری قرار می‌گیرد. در این حالت الگوهای خوشه‌بندی از بارکاری مشاهده شده، یادگیری می‌شوند. در پایان مرحله یادگیری، اطلاعات خوشه‌ها به بخش کشف تغییرات بارکاری ارسال می‌شوند. پس از آن، با مشاهده هر رویداد از جریان پیوسته رویدادها، فرایند خوشه‌بندی به حالت ثابت یا تطبیقی تغییر می‌کند. برای تضمین سازگاری، درحالی‌که پایش پیوسته و مبتنی بر جریان بارکاری حفظ شود، پس از مرحله یادگیری مراکز خوشه‌ها ثابت می‌شوند. بنابراین در حالت ثابت و تطبیقی هیچ تغییری در مراکز خوشه‌ها ایجاد نمی‌شود و ثابت هستند. اگر بردار ویژگی از رویداد جاری در مجموعه بردارهای ویژگی خوشه‌بندی‌شده وجود دارد، حالت ثابت فراخوانی می‌شود و در این حالت، رویداد مشاهده شده به خوشه‌ای که مرکز آن خوشه کمترین فاصله تا آن رویداد را دارا است، نسبت داده می‌شود. در غیر این صورت، با یک بردار ویژگی جدید روبرو هستیم که نشان‌دهنده یک نوع رویداد جدید است که باید خوشه‌ای برای آن ایجاد شود. این تصمیم‌گیری نیازمند ذخیره تمام بردارهای متمایز خوشه‌بندی‌شده است. فرایند خوشه‌بندی در حالت ثابت و تطبیقی باقی می‌ماند تا زمانی که یک تغییر بارکاری کشف و به‌موجب آن مرحله یادگیری مجدداً فراخوانی شود. در ادامه به توضیح جزئیات هر یک از الگوریتم‌های حالت یادگیری، ثابت و تطبیقی پرداخته‌ایم.

شکل۴ جزئیات الگوریتم حالت یادگیری را به‌وضوح نشان می‌دهد. این الگوریتم برای تعیین پایان مرحله یادگیری مرتباً نرخ بردارهای ویژگی جدید که به بارکاری اضافه می‌شوند را بررسی می‌نماید. برای این هدف، ابتدا این الگوریتم مرتباً بارکاری پایش شده در بازه‌های زمانی منظم r_i از مرحله پایش را دریافت می‌کند. همچنین حداقل مدت یادگیری (T) ورودی دیگر این الگوریتم است.

Algorithm 1. clustering Patterns Learning

Input: Minimum Learning Period $\boldsymbol{T}$; Regular Time Intervals $\boldsymbol{r_i}$.

// Learning State
1. $i = 0$
2. W= W U ObserveWorkload(r_i)
3. DV_i= Store_Distinct_PropertyVectors(W)
4. $t = t + r_i$
5. **repeat**
6. $i = i + 1$
7. $W = W$ U ObserveWorkload(r_i)
8. DV_i= Store_Distinct_PropertyVectors(W)
9. $t = t + r_i$
10. **until** $DV_i = DV_{i-1}$ *and* $t >= T$
11. K = Count_Distinct_PropertyVectors(W)
12. Workload K-means Clustering(W, K)

شکل ۴: الگوریتم یادگیری الگوهای خوشه‌بندی

در هر بازه، کل بردارهای ویژگی متمایز تا آن زمان، ذخیره می‌شود و مجموع زمان بازه‌ها (t) به دست می‌آید (خطوط ۲-۴). این عمل تا زمانی که حداقل مدت یادگیری طی شود و دیگر تغییری در مجموعه بردارهای ویژگی متمایز ذخیره شده، حاصل نشود تکرار خواهد شد (خطوط ۵-۱۰). در پایان مرحله یادگیری، الگوریتم K-means برای خوشه‌بندی بارکاری مشاهده شده، فراخوانی می‌شود و مراکز خوشه‌ها و وزن هر خوشه را به دست می‌آورد.

شکل ۵ جزئیات الگوریتم حالت ثابت و تطبیقی را نشان می‌دهد. در این الگوریتم، اگر بردار ویژگی از رویداد جاری در مجموعه بردارهای ویژگی خوشه‌بندی‌شده وجود دارد، حالت ثابت فراخوانی می‌شود. در این حالت، رویداد مشاهده شده به خوشه‌ای که مرکز آن خوشه کمترین فاصله تا آن رویداد را دارا است، نسبت داده می‌شود. همچنین به وزن آن خوشه یکی اضافه می‌شود (خطوط ۱-۵). در غیر این صورت، با یک بردار ویژگی جدید روبرو هستیم که نشان‌دهنده یک نوع رویداد جدید است که باید خوشه‌ای برای آن ایجاد شود (حالت تطبیقی). بنابراین خوشه جدید ایجاد شده و وزن آن برابر یک می‌شود. همچنین بردار ویژگی جدید در مجموعه بردارهای ویژگی متمایز ذخیره می‌شود (خطوط ۶-۱۱).

در نهایت خروجی حاصل شده از بخش خوشه‌بندی که شامل اطلاعات مراکز خوشه‌ها و وزن هر خوشه است به بخش «کشف تغییرات بارکاری» ارسال می‌شوند. در ادامه به راه‌حل ارائه شده برای کشف تغییرات بارکاری می‌پردازیم.

Algorithm 2. cluster Assignment and adaptation

Input: Statement Property Vector $\boldsymbol{v_{obs}}$; Cluster Medoids $\boldsymbol{M}$; Cluster Weights $\boldsymbol{CW}$; Distinct Property Vectors $\boldsymbol{DV}$.
Output: Medoid of Statement Cluster and its Weight.

// **Stable State**
1. ***if*** $Search(v_{obs}, DV)$ *is not null* ***then***
2. $m_{nearest} = Get\_Nearest\_Medoid(v_{obs}, \mathrm{M})$
3. $Assign(v_{obs}, m_{nearest})$
4. $CW_{m_{nearest}} = CW_{m_{nearest}} + 1$
5. **return** $m_{nearest}, CW_{m_{nearest}}$

// **Adapting State**
6. **else**
7. $m_{new} = Add\_Cluster(M, v_{obs})$
8. $Store\_StmtPropertyVector(v_{obs}, DV)$
9. $CW_{m_{new}} = 1$
10. **end if**
11. **return** $m_{new}, CW_{m_{new}}$

شکل ۵: الگوریتم حالت ثابت و تطبیقی از فرایند خوشه‌بندی

بارکاری پایگاه داده در یک سازمان می‌تواند در طی زمان ثابت باشد که در این صورت پایش بر تغییرات بارکاری یک سربار اضافی و غیرضروری است. ولی هنگامی‌که بارکاری با تغییرات چشمگیری همراه است، تطبیق شِمای طراحی شده با این تغییرات و تنظیم مجدد آن امری مهم و ضروری است. با کشف تغییرات بارکاری می‌توان تحلیل سبک‌وزن و پیوسته‌ای از بارکاری را فراهم نمود. در تحلیل سبک‌وزن از بارکاری فقط تغییرات مهم در بارکاری منجر به فراخوانی تنظیم مجدد شِمای پایگاه داده می‌شوند و در مقابل نوسانات و تغییرات جزیی در بارکاری مقاوم است.

بارکاری ثابت، بارکاری با توزیع پرس‌وجوی ثابت است و بارکاری متغیر، بارکاری با توزیع پرس‌وجویی است که در طی زمان تغییر می‌کند. تغییر در توزیع بارکاری منجر به ایجاد ترکیب‌های بارکاری[۲۹] مختلف می‌شود. در حقیقت تغییر در میزان فراوانی (تعداد تکرار) تراکنش‌های خواندن و نوشتن (تغییرات در ترکیب واکشی/به‌روزرسانی) منجر به ایجاد ترکیب‌های بارکاری مختلف می‌شود. هر یک از این ترکیب‌های بارکاری منجر به یک شِمای متفاوت در پایگاه داده می‌شوند. بنابراین هدف کشف تغییر در توزیع بارکاری است.

در بخش کشف تغییرات بارکاری، ابتدا توسط مؤلفه «ساخت مدل بارکاری[۳۰]» مدلی از بارکاری ایجاد می‌شود. این مدل باید قابلیت یادگیری و تطبیق را داشته باشد. برای مدل‌سازی بارکاری و کشف تغییرات، مدل‌های آماری نظیر مدل‌های مارکوف مخفی، مدل‌های زنجیره مارکوف و مدل‌های n-gram که مبتنی بر توزیع احتمال رویدادهای بارکاری هستند، مفید می‌باشند. در این تحقیق از مدل آماری n-gram برای ایجاد مدل بارکاری استفاده می‌شود. مدل n-gram مبتنی بر زنجیره مارکوف است که بر طبق یک فرایند احتمالی، توالی از رویدادها را تولید می‌کند. یک مدل n-gram، مدل مارکوفی با مرتبه (n-1) است که ویژگی مارکوف را فقط به‌عنوان یک تقریب در نظر می‌گیرد. ویژگی مارکوف بیان می‌کند که احتمال یک رویداد فقط وابسته به تاریخچه‌ای از طول n رویداد قبلی است نه کل رویدادهای قبلی [۴۱]. مدل‌سازی بارکاری با زنجیره مارکوف نیازمند دانش کاملی از پردازش داخلی برنامه است. ولی ازآنجایی‌که مدل‌های n-gram، ویژگی مارکوف را فقط به‌عنوان یک تقریب در نظر می‌گیرند، برخلاف مدل‌های مارکوف مخفی و مدل‌های زنجیره مارکوف به‌طور خودکار قابل یادگیری و تطبیق هستند. بنابراین مدل‌های n-gram برای مدل‌سازی بارکاری و کشف تغییرات مناسب‌تر هستند.

برای یک بارکاری $W=(w_1, w_2, \ldots, w_t)$ که توالی از خوشه‌های بارکاری است، هر خوشه w_t در مدل n-gram (زنجیره مارکوفی به طول n-1) به یک حالت نگاشت می‌شود. در حقیقت نگاشت بارکاری W به حالات مدل، بر مبنای نوع جملات پرس‌وجو است، بدین‌صورت که هر نوع جمله به یک حالت در مدل نگاشت می‌شود و احتمال انتقال بر طبق ترتیب جملات در بارکاری محاسبه می‌شود. احتمال w_t با رابطه (۱) محاسبه می‌شود.

$$P(w_t \mid w_{t-(n-1)} \cdots w_{t-1}) \cong \frac{P(w_t, w_{t-(n-1)}, \ldots, w_{t-1})}{P(w_{t-(n-1)}, \ldots, w_{t-1})} = \frac{Count(w_t, w_{t-(n-1)}, \ldots, w_{t-1})}{Count(w_{t-(n-1)}, \ldots, w_{t-1})} \quad (۱)$$

سپس توسط مؤلفه «ارزیابی و تطبیق بارکاری[۳۱]» بارکاری جاری با مدل ایجاد شده مرتباً مقایسه می‌شود و درصورتی‌که با مدل جاری همانند نباشد و دچار تغییرات شده باشد، تطبیق مدل با تغییرات ایجاد شده در بارکاری انجام می‌شود. همچنین در این زمان منطق خودتنظیمی از مرحله طرح‌ریزی برای انتخاب شِمای مناسب (تنظیم مجدد شِما)، رهانا می‌شود.

در مؤلفه «ارزیابی و تطبیق بارکاری» برای کشف تغییرات باید بارکاری جاری با مدل بارکاری ایجاد شده مقایسه شود، بنابراین باید یک معیار شباهت مناسب انتخاب کرد. یکی از معیارهایی که برای ارزیابی کیفیت مدل آماری n-gram مورداستفاده قرار می‌گیرد، معیار سرگشتگی[۳۲] (PP) است[۴۱]. این معیار می‌تواند میزان شباهت بین مدل بارکاری و بارکاری جاری را اندازه‌گیری کند. مقادیر پایین برای این معیار نشان می‌دهد که بارکاری جاری به‌خوبی با مدل همانند است و مقادیر بالا نشان‌دهنده وجود یک تغییر چشمگیر در بارکاری جاری است. معیار PP برای بارکاری W از مدل آماری n-gram از رابطه (۲) به دست می‌آید.

$$PP(W) = \hat{P}(w_1 w_2 w_3 \cdots w_T)^{-1/T} = \left(\prod_{t=1}^{T} P(w_t \mid w_{t-(n-1)} \cdots w_{t-1})\right)^{-1/T} \quad (۲)$$

شکل ۶ مراحل الگوریتم پیشنهادی برای کشف تغییرات بارکاری با استفاده از مدل n-gram را نشان می‌دهد. مطابق شکل ۳ منطق کشف تغییرات ارائه شده، شامل دو حالت یادگیری و تطبیقی است. ابتدا فرایند کشف تغییرات در حالت اولیه یادگیری قرار می‌گیرد که

خوشه‌های بارکاری جاری را به مدل n-gram تبدیل می‌کند. احتمال انتقال حالات در مدل توسط رابطه (۱) محاسبه می‌شود. سپس با استفاده از رابطه (۲) معیار PP برای بارکاری جاری محاسبه می‌شود. اگر مقدار این معیار بیشتر از حد آستانه تعیین شده باشد، نشان‌دهنده وجود یک تغییر در بارکاری جاری است. ازاین‌رو فرایند کشف تغییر به حالت تطبیقی تغییر می‌کند و مدل بارکاری با تغییرات تطبیق داده می‌شود و منطق خودتنظیمی از مرحله طرح‌ریزی برای انتخاب شِمای مناسب (تنظیم مجدد شِما)، رهانا می‌شود.

Algorithm 3. Workload Change Detection Algorithm

Input: Workload $\boldsymbol{W_{obs}}$; Workload Model $\boldsymbol{M}$; The User-specified Perplexity Threshold $\boldsymbol{T}$.

// **Learning State**
1. **If** WorkloadChangeDetectionLogic.state = $learning$ **then**
2. nGram_model=ConvertTonGram(W_{obs} , $Chain\_Lenght$)
3. **end if**
4. PP = M.ComputePerplexity(nGram_model)
5. **If** $PP > T$ **then**
6. WorkloadChangeDetectionLogic.state = $adapting$
7. **end if**

// **Adapting State**
8. **If** WorkloadChangeDetectionLogic.state = $adapting$ **then**
9. $M.AdaptFrom$(nGram_model)
10. RaiseAlert (“workload Change Detection.”)
11. $Trigger.SchemaDesignTuning()$
12. **end if**

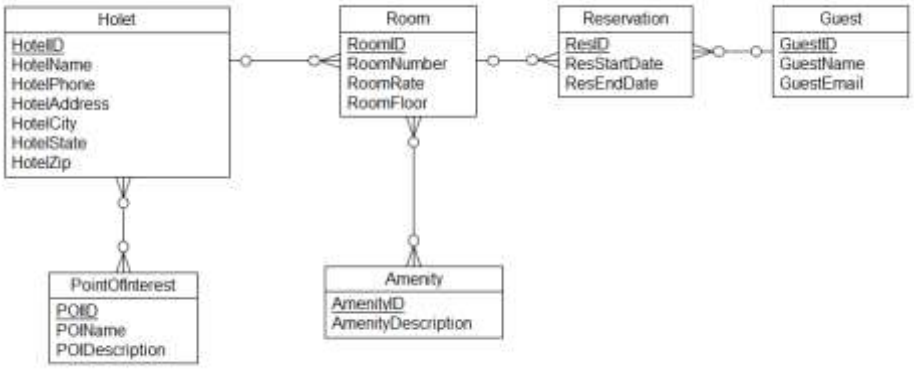
شکل ۶: الگوریتم کشف تغییرات بارکاری

۵ -نتایج

در این بخش ابتدا با ارائه مثالی ساده مراحل راهکار پیشنهادی برای کشف تغییرات بارکاری در پایگاه داده ستون‌گرا تشریح می‌گردد. سپس راهکاری پیشنهادی در سطح بالای پایگاه داده کاساندرا پیاده‌سازی و با استفاده از مطالعه موردی سیستم‌های تجارت الکترونیک تحت وب ارزیابی می‌شود.

۵ ۱ -مثالی از کشف تغییرات بارکاری

با استفاده از مثال سیستم رزرو هتل که برگرفته از کتاب راهنمای کامل کاساندرا [۳] است، به چگونگی کشف تغییرات بارکاری در پایگاه داده ستون‌گرا با استفاده از راهکار پیشنهادی می‌پردازیم. شکل ۷ مدل مفهومی پایگاه داده را برای این سیستم نمایش می‌دهد.



شکل ۷: مدل مفهومی سیستم رزرو هتل

ابتدا فرض می‌کنیم که بارکاری برنامه هتل، شامل پرس‌وجوهای زیر باشد:

۱. به دست آوردن مکان‌های تفریحی نزدیک هتل‌هایی که توسط یک مسافر رزرو شده است.
۲. به دست آوردن مکان‌های تفریحی نزدیک یک هتل مشخص
۳. به دست آوردن نرخ و تسهیلات اتاق‌های یک هتل مشخص
۴. رزرو هتل توسط مسافر

شکل ۸ مجموعه گروه‌های ستون(CF) که باید برای پاسخ به پرس‌وجوهای مذکور ایجاد شوند را نشان می‌دهد.اگر بارکاری برنامه تغییر کند و شامل به‌روزرسانی مکرر نام و توصیف مکان‌های تفریحی باشد، CF های نشان داده در شکل ۸ نمی‌توانند مطلوب باشند. زیرا اطلاعات مکان‌های تفریحی غیرنرمال سازی شده‌اند: نام و توصیف یک مکان تفریحی ممکن است در بسیاری از رکوردهای هتل‌ها ذخیره شده باشد. بنابراین بهتر است که بجای CF_2 از دو CF_{2a} و CF_{2b} ایجاد شود (شکل ۹).

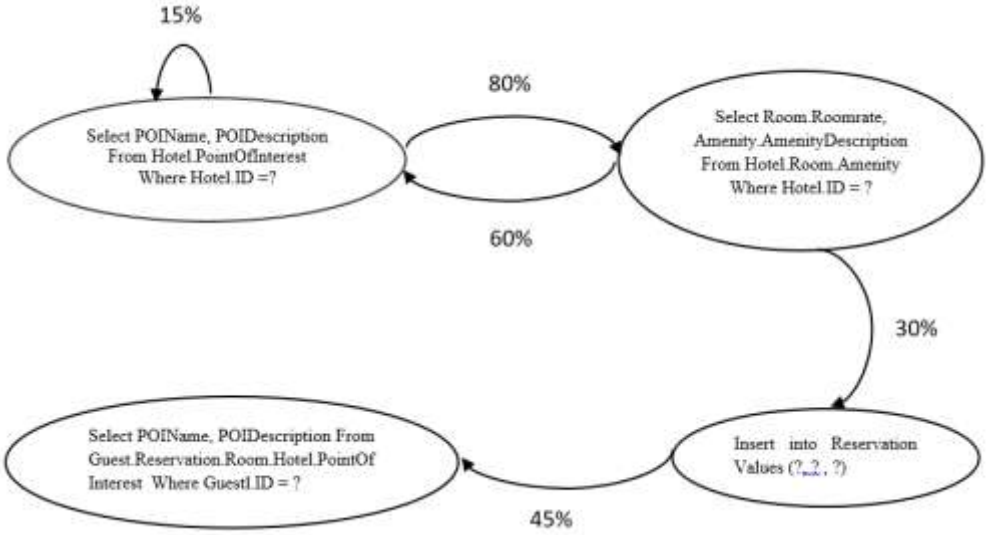
CF_1: [GuestID] [HotelID] []
CF_2: [HotelID] [POIID] [POIname, POIDescription]
CF_3: [HotelID] [RoomID, AmenityID]
[RoomNumber, RoomRate, RoomFloor, AmenityDescription]
CF_4: [ResID] [] [GuestID, HotelID, RoomID]

شکل ۸: مجموعه CF ها برای بارکاری هتل

CF_4: [ResID] [] [GuestID, HotelID, RoomID]
CF_1: [GuestID] [HotelID] []
CF_{2a}: [HotelID] [POIID] []
CF_{2b}: [POIID] [] [POIname, POIDescription]
CF_3: [HotelID] [RoomID, AmenityID]
[RoomNumber, RoomRate, RoomFloor, AmenityDescription]
CF_4: [ResID] [] [GuestID, HotelID, RoomID]

شکل ۹: مجموعه CF های جدید با تغییر در توزیع بارکاری هتل

اگر در بارکاری ورودی نرخ واکشی اطلاعات مکان‌های تفریحی کم و نرخ به‌روزرسانی آن‌ها بالا باشد، شمای دوم مناسب است. ولی اگر نرخ واکشی اطلاعات مکان‌های تفریحی زیاد و نرخ به‌روزرسانی آن‌ها کم باشد، شمای اول مناسب است. بنابراین تغییر در توزیع‌های بارکاری، منجر به شِمای متفاوتی می‌شود. برای کشف این تغییر باید با استفاده از مدل n-gram، مدلی از بارکاری اولیه برنامه ایجاد شود. شکل ۱۰ مدل bigram (زنجیره مارکوف به طول یک) از بارکاری اولیه را نشان می‌دهد. سپس با پایش پیوسته بارکاری جاری و مقایسه آن با مدل ساخته شد با استفاده از معیار سرگشتگی، این تغییر کشف و تغییرات بارکاری به مدل ایجاد شده اعمال می‌شود. در نهایت طراحی مجدد شِما از مرحله طرح‌ریزی فراخوانی می‌شود که منجر به ایجاد مجموعه CF ها نشان داده شده در شکل ۹ می‌شود.



شکل ۱۰: مدل bigram (زنجیره مارکوف به طول یک) از بارکاری هتل

نمونه‌ای از راهکار پیشنهادی در این سیستم پایگاه داده پیاده‌سازی و ارزیابی خواهد شد. مطالعه موردی RUBIS یک برنامه محک تحت وب است که عملکرد یک سایت حراجی مانند ebay را شبیه‌سازی می‌کند. تعاملات اصلی کاربران در این سایت حراج شامل فروش، جستجو و پیشنهاد قیمت است. مدل مفهومی RUBIS شامل هفت موجودیت کاربران، کالاها، دسته‌ها، نواحی، مبلغ پیشنهادی، خرید فوری و نظرات و ارتباط‌های بین آن‌ها است (شکل ۷).

۵ ۴ -ارزیابی

با توجه به تمرکزی که مطالعات حوزه خودکارسازی طراحی شِمای پایگاه داده NoSQL در این اواخر داشته‌اند، مورد مطالعه سیستم‌های تجارت الکترونیک تحت وب نظیر وب‌سایت حراجی RUBIS[۳۳] یا کتاب‌فروشی برخط TPC-W مناسب به نظر می‌رسد. همچنین برای ارزیابی روش پیشنهادی در کشف تغییرات باید سیستم‌هایی انتخاب شوند که دارای ترکیب‌های مختلف بارکاری باشند تا بتوان کارایی روش پیشنهادی را در کشف تغییرات ارزیابی نمود. سایت حراجی RUBIS دارای ترکیب‌های مختلف بارکاری نظیر browsing و bidding است و چنین امکانی را فراهم می‌کند. بنابراین در این بخش به ارزیابی راهکار پیشنهادی با استفاده از مطالعه موردی سیستم‌های تجارت الکترونیک تحت وب نظیر وب‌سایت حراجی RUBIS پرداخته می‌شود. این ارزیابی شامل ارزیابی کشف تغییرات در بارکاری و سربار حاصل از آن است.

در این تحقیق سیستم پایگاه داده کاساندرا که یک پایگاه داده ستون‌گرا NoSQL است برای پیاده‌سازی و ارزیابی راهکار پیشنهادی در نظر گرفته شده است. برای تطبیق پایگاه داده رابطه‌ای RUBIS با کاساندرا، یک مدل مفهومی بر مبنای موجودیت‌های آن ایجاد و سپس

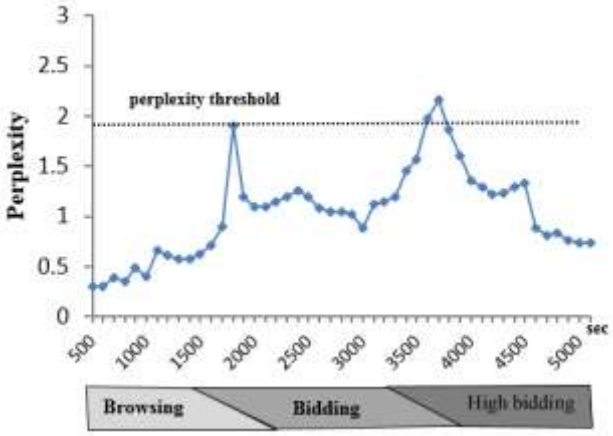
تعاملات وب RUBIS به دو دسته browsing و bidding تقسیم می‌شوند. تعاملات browsing فقط عملیات خواندن از پایگاه داده را انجام می‌دهند، درحالی‌که تعاملات bidding عملیات خواندن از و نوشتن در پایگاه داده را انجام می‌دهند. بنابراین پایگاه داده سایت حراجی RUBIS شامل دو ترکیب بارکاری مختلف است. ترکیب بارکاری browsing فقط شامل تعاملات خواندن از پایگاه داده است و ترکیب بارکاری bidding شامل ۸۵٪ تعاملات فقط خواندنی و ۱۵٪ تعاملاتی که شامل نوشتن در پایگاه داده است. در بارکاری bidding با افزایش وزن تعاملاتی که شامل نوشتن هستند می‌توان توزیع بارکاری

جدیدی را تعریف کرد. در این تحقیق برای آزمایش و ارزیابی راهکار پیشنهادی، ترکیب بارکاری دیگری نظیر high bidding تعریف می‌شود که نرخ تعاملات bidding در آن بیشتر است. ترکیب‌های مختلف بارکاری از RUBIS در جدول ۲ نشان داده شده است.

جدول ۲: ترکیب‌های مختلف بارکاری از RUBIS

Interaction	Browsing	Bidding	High bidding
Browse	100%	85%	60%
Bid	0%	15%	40%

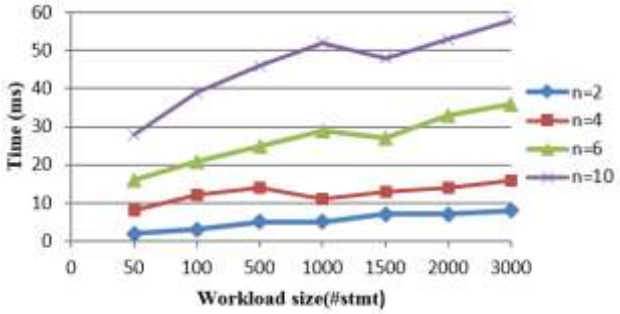
همان‌طورکه گفته شد، در این پژوهش برای مدل‌سازی بارکاری و کشف تغییرات از مدل آماری n-gram که مبتنی بر توزیع احتمال رویدادهای بارکاری است، استفاده شده است. بنابراین در این قسمت به ارزیابی کیفیت مدل n-gram ایجاد شده از بارکاری می‌پردازیم. بدین منظور از معیار سرگشتگی(*PP*) استفاده می‌شود. همان‌طورکه در بخش چهارم بیان شد، این معیار میزان شباهت بین مدل بارکاری و بارکاری جاری را اندازه‌گیری می‌کند. مقادیر پایین برای این معیار نشان می‌دهد که بارکاری جاری به‌خوبی با مدل همانند است و مقادیر بالا نشان‌دهنده وجود یک تغییر چشمگیر در بارکاری جاری است. مدل n-gram قوی‌تر، سرگشتگی کمتری را نتیجه می‌دهد. مقدار سرگشتگی از رابطه (۲) که در بخش چهارم آورده شده است، محاسبه می‌شود. شکل ۱۱ مقادیر *PP* را در ترکیب‌های مختلف بارکاری RUBIS نشان می‌دهد. همان‌طورکه مشاهده می‌شود معیار *PP* در زمانی که بارکاری ثابت است مقادیر پایین و در نقاطی که ترکیب بارکاری تغییر کرده است، مقادیر بالا را نشان می‌دهد. بنابراین راهکار ارائه شده به‌خوبی می‌تواند تغییر در ترکیب‌های مختلف بارکاری را کشف کند.



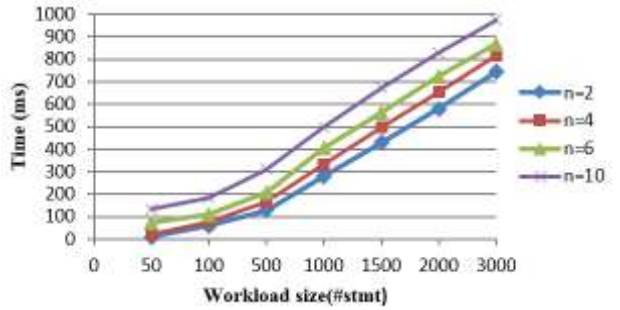
شکل ۱۱: مقادیر معیار سرگشتگی(*PP*) برای ترکیب‌های مختلف بارکاری RUBIS

سربار حاصل از کشف تغییرات بارکاری برابر با زمان محاسبه معیار شباهت *PP* و زمان تطبیق مدل با تغییرات است که وابسته به طول زنجیره مارکوف و سایز بارکاری است.

شکل ۱۲ زمان محاسبه معیار شباهت *PP* و شکل ۱۳ زمان تطبیق مدل با تغییرات را برای مقادیر مختلف n در مدل n-gram نشان می‌دهند. همان‌طورکه نتایج نشان می‌دهد، روش پیشنهادی کشف تغییرات برای بارکاری ثابت، سربار بسیار کمی را دارد و کارا است. زیرا زمان محاسبه معیار سرگشتگی حتی برای مقادیر بالای n کمتر از ۶۰ms است.



شکل ۱۲: محاسبه معیار سرگشتگی (PP)



شکل ۱۳: محاسبه زمان تطبیق مدل با تغییرات

۶ -نتیجه‌گیری و کارهای آتی

در این مقاله یک حلقه کنترل بازخورد برای پایش پیوسته و تحلیل سبک وزن در پایگاه داده ستون‌گرا NoSQL ارائه شد. این حلقه توصیف کننده یک الگوی طراحی برای ویژگی خودتنظیمی است و برای کشف تغییرات بارکاری بکار می‌رود که لازمه بازتنظیمی خودکار شِمای پایگاه داده است. در این حلقه مرحله پایش به‌طور پیوسته به مشاهده جملات پرس‌وجوی بارکاری و استخراج ویژگی‌های مرتبط با آن‌ها می‌پردازد. سپس بردارهای ویژگی حاصل شده به مرحله تحلیل ارسال می‌شوند. در مرحله تحلیل به خوشه‌بندی رویدادهای بارکاری و کشف تغییرات حاصل شده در آن در طی زمان پرداخته شده است. با کشف تغییرات از مرحله تحلیل، منطق خودتنظیمی برای طراحی مجدد شِما و انتخاب شِمایی متناسب و بهینه با تغییرات بارکاری از

مرحله طرح‌ریزی رهانا می‌شود. همان‌طورکه گفته شد، تمرکز اصلی این پژوهش بر روی مراحل پایش و تحلیل برای کشف تغییرات در بارکاری است. نتایج به‌دست‌آمده از پیاده‌سازی و اجرای راهکار پیشنهادی، کارایی آن را در کشف تغییرات اثبات نمود.

از کارهای آینده، پیشنهاد ابزاری برای انتخاب خودکار شِمای بهینه در مرحله طرح‌ریزی می‌باشد. همچنین یکی دیگر از کارهای آتی شامل تحلیل مدل مفهومی برای کشف تغییرات در آن و تطبیق شِمای پایگاه داده NoSQL با تغییرات کشف شده است.

مراجع

[1] R. A. Nzekwa, R. Rouvoy, and L. Seinturier, "Modelling feedback control loops for self-adaptive systems", Electronic Communications of the EASST, vol. 28, no. 2, pp. 106-121, 2010.

[2] S. Chaudhuri and V. Narasayya, "Self-tuning database systems: a decade of progress", in Proceedings of the 33rd international conference on Very large data bases, pp. 3-14, 2007.

[3] E. Hewitt, *Cassandra: the definitive guide*, O'Reilly Media, 2010.

[4] M. J. Mior, K. Salem, A. Aboulnaga, and R. Liu, "NoSE: Schema design for NoSQL applications", in Data Engineering (ICDE), 2016 IEEE 32nd International Conference on, pp. 181-192. 2016

[5] D. Bermbach, S. Müller, J. Eberhardt, and S. Tai, "Informed Schema Design for Column Store-Based Database Services", in Service-Oriented Computing and Applications (SOCA), IEEE 8th International Conference on, pp. 163-172, 2015.

[6] M. Boussahoua, O. Boussaid, and F. Bentayeb, "Logical Schema for Data Warehouse on Column-Oriented NoSQL Databases", in International Conference on Database and Expert Systems Applications, pp. 247-256, 2017.

[7] A. Chebotko, A. Kashlev, and S. Lu, "A big data modeling methodology for Apache Cassandra", in Big Data (BigData Congress) 2015 IEEE International Congress on , pp. 238-245,2015.

[8] C. de Lima and R. dos Santos Mello, "A workload-driven logical design approach for NoSQL document databases", in Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services, p. 73-79, 2015.

[9] M. J. Mior, K. Salem, A. Aboulnaga, and R. Liu, "NoSE: Schema design for NoSQL applications", IEEE Transactions on Knowledge and Data Engineering, vol. 29, no. 10, 2017.

[10] T. Vajk, L. Deák, K. Fekete, and G. Mezei, "Automatic nosql schema development: A case study", in Artificial Intelligence and Applications, pp. 656-663, 2013.

[11] T. Vajk, P. Feher, K. Fekete, and H. Charaf, "Denormalizing data into schema-free databases", in Cognitive Infocommunications (CogInfoCom), 2013 IEEE 4th International Conference on, pp. 747-752: IEEE 2013.

[12] F. Yang, D. Milosevic, and J. Cao ,"Optimising column family for OLAP queries in HBase", International Journal of Big Data Intelligence, vol. 4, no. 1, pp. 23-35, 2017.

[13] G. Valentin, M. Zuliani, D. C. Zilio, G. Lohman, and A. Skelley, "DB2 advisor :An optimizer smart enough to recommend its own indexes", in Data Engineering, 2000. Proceedings. 16th International Conference on, pp. 101-110: IEEE 2000.

[14] D. C. Zilio, J. Roa, S. Lightstone, G, Lohman, A. Storm, and S. Fadden, "DB2 design advisor: integrated automatic physical database design", in Proceedings of the Thirtieth international conference on Very large data bases-Volume 30, pp. 1087-1097: VLDB Endowment 2004.

[15] B. Dageville, D. Das, K. Dias, K. Yagoub, M. Zait, and M. Ziauddin, "Automatic SQL tuning in Oracle 10g", in Proceedings of the Thirtieth international conference on Very large data bases-Volume 30, pp. 1098-1109: VLDB Endowment 2004.

[16] S. Agrawal, S. Chaudhuri, L. Kollar, A. Marathe, V. Narasayya, and M. Syamala, "Database tuning advisor for Microsoft SQL Server 2005: demo" ,in Proceedings of the 2005 ACM SIGMOD international conference on Management of data, pp. 930-932: ACM 2005.

[17] N. Bruno and S. Chaudhuri, "An online approach to physical design tuning", in Data Engineering, ICDE 2007. IEEE 23rd International Conference on, pp. 826-835: IEEE 2007.

[18] M. Holze and N. Ritter, "Towards workload shift detection and prediction for autonomic databases", in Proceedings of the ACM first Ph. D .workshop in CIKM, pp. 109-116: ACM 2007.

[19] M. Holze and N. Ritter, "Autonomic databases: Detection of workload shifts with n-gram-models", in East European Conference on Advances in Databases and Information Systems, pp. 127-142: Springer, Berlin, Heidelberg 2008.

[20] K. Schnaitter, S. Abiteboul, T. Milo, and N. Polyzotis, "Colt: continuous on-line tuning", in Proceedings of the 2006 ACM SIGMOD international conference on Management of data, pp. 793-795: ACM 2006.

[21] K. Schnaitter, S. Abiteboul, T. Milo, and N. Polyzotis, "On-line index selection for shifting workloads", in Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop, pp. 459-468: IEEE Computer Society, 2007.

[۲۲] محیا ارومیه و نگین دانش‌پور، «مدلی سه لایه در طراحی سطح منطقی پایگاه داده تحلیلی»، مجله مهندسی برق، دانشگاه تبریز، جلد ۴۷، شماره ۲، صفحات ۳۷۱-۳۸۰، ۱۳۹۶.

[۲۳] پروانه شایق بروجی و نگین دانشپور، « انتخاب دید جهت ذخیرهسازی دید در پایگاه داده تحلیلی با استفاده از الگوریتم فرهنگی ترکیبی»، مجله مهندسی برق، دانشگاه تبریز، جلد ۴۶، شماره ۲، صفحات ۹۷-۱۰۸، ۱۳۹۵.

[24] A. Pavlo *et al.*, "Self-Driving Database Management Systems" ,in CIDR 2017, Conference on Innovative Data Systems Research, January 8-11, Chaminade, *CA*, 2017.

[25] R. Schroeder and R. d. S. Mello, "Improving query performance on XML documents: a workload-driven design approach", in Proceedings of the eighth ACM symposium on Document engineering, pp. 177-186: ACM 2008.

[26] P. S. Yu, M.-S. Chen, H.-U. Heiss, and S. Lee, "On workload characterization of relational database environments", IEEE Transactions on Software Engineering, vol. 18, no. 4, pp. 347-355, 1992.

[27] S. Elnaffar, P. Martin, B. Schiefer, and S. Lightstone, "Is it DSS or OLTP: automatically identifying DBMS workloads", Journal of Intelligent Information Systems, vol. 30, no. 3, pp. 249-271, 2008.
[28] S. Elnaffar and P. Martin, "The Psychic–Skeptic Prediction framework for effective monitoring of DBMS workloads", Data & Knowledge Engineering, vol. 68, no. 4, pp. 393-414, 2009.
[29] Z. Zewdu, M. K. Denko, and M. Libsie, "Workload characterization of autonomic dbmss using statistical and data mining techniques", in Advanced Information Networking and Applications Workshops, WAINA'09. International Conference on, pp. 244-249: IEEE 2009.
[30] M. Holze and N. Ritter, "Autonomic Databases: Detection of Workload Shifts with n-Gram-Models", In East European Conference on Advances in Databases and Information Systems (pp. 127-142). Springer, Berlin, Heidelberg, 2008.
[31] Q. Yao, A. An, and X. Huang, "Finding and analyzing database user sessions", In International Conference on Database Systems for Advanced Applications (pp. 851-862). Springer, Berlin, Heidelberg, 2005.
[32] M. Abdul, A. M. Muhammad, N. Mustapha, S. Muhammad, and N. Ahmad, "Database workload management through CBR and fuzzy based characterization", Applied Soft Computing, vol. 22, pp. 605-621, 2014.
[33] A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches", AI communications, vol. 7, no. 1 , pp. 39-59, 1994.
[34] F. Bugiotti, L. Cabibbo, P. Atzeni, and R. Torlone, "Database design for NoSQL systems", in International Conference on Conceptual Modeling, pp. 223-231: Springer 2014.
[35] M. C. Huebscher and J. A. McCann, "A survey of autonomic computing—degrees, models, and applications", ACM Computing Surveys (CSUR), vol. 40, no. 3, pp. 191-213, 2008.
[36] W. Karwowski, *International encyclopedia of ergonomics and human factors*, Second Edition ed. Crc Press, 2006.
[37] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.
[38] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*. Boston: Pearson Addison Wesley, 2005.
[39] C. Ordonez and E. Omiecinski, "FREM: fast and robust EM clustering for large data sets", in Proceedings of the eleventh international conference on Information and knowledge management, pp. 590-599: ACM 2002.
[40] L. O'callaghan, N. Mishra, A. Meyerson, S. Guha, and R. Motwani, "Streaming-data algorithms for high-quality clustering, in Data Engineering", Proceedings 18th International Conference on, pp.685-694, 2002.
[41] D. Jurafsky and J. H. Martin, *Speech and language processing*. Pearson London, 2014.

زیرنویس‌ها

[1] Self-adaptive
[2] Autonomic Database
[3] Self-management
[4] Feedback Control Loop
[5] Self-Tuning
[6] Automated Schema Database Tuning
[7] Automated Physical Database Design
[8] Workload
[9] Wide Column Store
[10] Column Families
[11] Advisory Tools
[12] Self-driving
[13] Peloton Database Management System, 2016, http://pelotondb.org.
[14] NoSQL Schema Evaluator
[15] binary integer program
[16] RElational Database Workload AnalyzeR
[17] Benchmark Workload
[18] Decision Support System
[19] Psychic-Skeptic Prediction
[20] Classification and Regression Tree
[21] Case-Based Reasoning
[22] Workload Monitoring
[23] Event-based Monitoring
[24] Time-based Monitoring
[25] Workload Characterization
[26] Workload Self-adaptive & Dynamic Clustering
[27] Workload Change Detection
[28] Well-Separated
[29] Workload Mixes
[30] Workload Model Construction
[31] Workload Evaluation & Adaptation
[32] Perplexity
[33] RUBIS: Rice University bidding system. http://rubis.objectweb.org