ISSN: 2745-3673
http://www.Rcsj.ir

مجله‌ی علمی پژوهش در علوم رایانه (سال دوم)
شماره ۶ / تابستان ۱۳۹۶ / ص۲۵- ۱۴

الگویNoSQL برای مدیریت و مفاهیم پایگاه داده ها در کاساندرا

طاهره حجری[۱] جعفر پرتابیان[۲]

[۱] دانشجوی کارشناسی ارشد، دانشگاه آزاد اسلامی واحد لامرد

[۲] مدیرگروه کامپیوتر، دانشگاه ازاد اسلامی واحد لامرد

نام نویسنده مسئول:
طاهره حجری

چکیده

در سال ۲۰۰۸ دو کارمند سابق آمازون و مایکروسافت که در یک شبکه اجتماعی بزرگ مشغول به کار بودند، یک پایگاه داده جدید **No SQL** را ایجاد کردند که کاساندرا نامیده شد. کاساندرا شایستگی‌های بسیاری همچون ماندگاری طولانی، مقیاس‌پذیری بسیار بالا و ثبات تنظیم‌پذیر دارد و به همین دلیل، توسعه‌دهندگان بزرگی از طرف شرکت‌های معتبر در توسعه کد آن سهیم هستند. کاساندرا یک سیستم ذخیره سازی توزیع شده برای مدیریت مقادیر بسیار زیادی از داده های ساخت یافته در سراسر بسیاری از سرور های کالا گسترش یافته است، در حالی که ارائه خدمات بسیار در دسترس دارد. نوشتار حاضر مفاهیم کلی مطرح در کاساندرا، روش کار و نحوه برقراری ارتباط با آن از طریق برنامه‌های کاربردی**NET**. را مرور خواهد کرد

واژگان کلیدی: پایگاه داده ی رابطه ای ، پایگاه داده ی غیررابطه ای ، مقیاس‌پذیری ،کاساندرا، ذخیره سازی.

مجلـه‌ی علمی پژوهش در علوم رایانه ، شماره ۶ ، تابستان ۱۳۹۶، ص ۲۵- ۱۴

ISSN: 2745-3673
http://www.Rcsj.ir

مقدمه

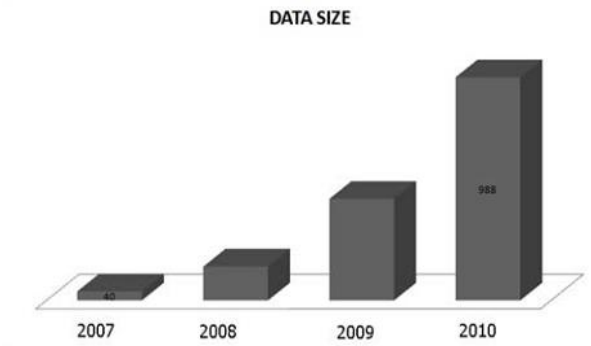
به دلیل پیشرفت های اخیر در کاربرد داده های توزیع شده و مشکلات پایگاه داده های رابطه ای که فاقد رسیدگی به رشد نمایی داده ها هستند، نیاز به ذخیره سازی های انبوه داده ها با قابلیت مقیاس پذیری و عدم محدودیت در الگوها ایجاد شده (Swamy,۲۰۱۳)است.بسیاری از سازمان ها اطلاعات مورد نیاز خود را برای تجزیه و تحلیل در آینده جمع آوری می کنند. به طور معمول بسیاری از آن ها، داده های ساختار یافته را برای دسترسی های بعدی و تجزیه و تحلیل در پایگاه داده های رابطه ای ذخیره می کنند. با این حال تعداد فزاینده ای از سازمان ها و تولیدکنندگان ، پایگاه داده رابطه ای خود را به یکی از انواع مختلف پایگاه داده های غیررابطه ای که در حال حاضر، پایگاه داده NoSqlنامیده می شوند، تغییر داده اند. مزیت اصلی این نوع پایگاه داده ها این است که برخلاف پایگاه داده های رابطه ای ، آنها به داده های غیر ساختار یافته نظیر اسناد نامه های الکترونیکی و چندرسانه ای به طور مؤثر رسیدگی می کنند. ویژگی های مشترک پایگاه داده های NoSqlمی توانند در قابلیت گسترش ، مقیاس پذیری بالا، قابلیت اطمینان ، مدل داده ای و زبان پرس وجوی بسیار ساده ، عدم وجود مکانیزمی برای اداره کردن و مدیریت سازگاری داده و حفظ محدودیت های تمامیت خلاصه گردند (۲۰۱۲ ., Taura et)پایگاه داده های مختلف NoSql، رویکردهای متفاوتی دارند. یک نمونه قوی از پایگاه داده های NoSql،کاساندرا نامیده شده است که در ابتدا برای استفاده در فیس بوک توسعه یافت (Swamy, ۲۰۱۳). فیس بوک بزرگترین پلت فرم شبکه های اجتماعی است که صدها میلیون کاربر در زمان اوج استفاده از آن ، از ده ها هزار سرور که در بسیاری از مراکز داده در سراسر جهان قرار گرفته اند، خدمات می گیرند. نیازمندی های عملیاتی جدی در پلت فرم فیس بوک از لحاظ عملکرد، قابلیت اطمینان و بهره وری وجود دارند و برای حمایت از رشد مداوم پلت فرم ، نیاز به قابلیت گسترش و برخورد با شکست ها در یک زیر ساخت متشکل از هزاران جزء، می باشد. برای پاسخگویی به نیازهای قابلیت اطمینان و مقیاس پذیری در فیس بوک ،کاساندرا توسعه داده شد (Lakshman and Malik, ۲۰۱۰).

در ادامه مقاله به این صورت سازماندهی می گردد: در رابطه با کارهای مرتبط می باشد که برخی از آنها در طراحی کاساندرا بسیار مؤثر بوده اند. مشکلات مربوط به پایگاه داده های سنتی رابطه ای مطرح می گردند. ضمن معرفی اجمالی پایگاه داده NoSql، طبقه بندی آن و مزایا و معایب استفاده از آن بیان خواهند شد. پس از آشنایی با پایگاه داده کاساندرا به عنوان یک نمونه از پایگاه داده های غیررابطه ای ، مدل داده ، معماری ، ویژگی های اساسی ، زمینه های استفاده و یک نمونه پیاده سازی عملی آن شرح داده می شوند. نتیجه گیری و کارهای آتی نیز در بخش نهایی مورد بحث قرار می گیرند.

مشکلات پایگاه داده ی رابطه ای

سه مشکل اصلی در رابطه با پایگاه داده ی رابطه ای وجود دارد که آن را ناکارآمد می سازند، در ادامه هر کدام از این مشکلات بیان خواهند شد.

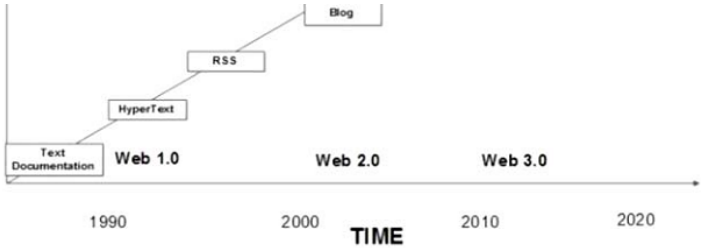
اولین مشکل ، اندازه ی مجموعه داده است . رشد عظیمی از اطلاعات در اینترنت وجود دارند. مطابق تجزیه و تحلیل مؤسسه ی بین المللی داده که در شکل (۱) نشان داده شده است ، بیان می شود که رشد داده از سال ۲۰۰۷ تا سال ۲۰۱۰ حدوداً ۲۵ برابر شده است (Taura et al., ۲۰۱۲).



شکل (۱) رشد داده از سال ۲۰۰۷ تا سال ۲۰۱۰ (Taura et al., ۲۰۱۲)

دومین مشکل ، قابلیت ارتباط است . با گذشت زمان اطلاعات بیشتر به یکدیگر مرتبط می شوند. شکل (۲) رشد قابلیت ارتباط را در طول چندین سال نشان می دهد (Taura et al., ۲۰۱۲).

ISSN: 2745-3673
http://www.Rcsj.ir



شکل (۲) رشد قابلیت ارتباط اطلاعات (Taura et al., ۲۰۱۲)

سومین مشکل در ارتباط با اطلاعات نیمه ساختاریافته است . اطلاعات نیمه ساختار یافته ، اطلاعاتی هستند که تعداد کمی ویژگی اجباری دارند، در حالی که تعداد زیادی ویژگی اختیاری دارند. رشد اطلاعات نیاز به افزایش ستون های جداول را به وجود می آورد، که این تغییر منجر به پراکندگی جداول می گردد (Taura et al., ۲۰۱۲).

علاوه مشکلات فوق الذکر، برای پایگاه داده های رابطه ای می توان مشکلات دیگری از جمله : عدم حمایت از همروندی بالای خواندن و نوشتن با تأخیر کم ، عدم توانایی در ذخیره سازی کارآمد داده های بزرگ ، محدودیت قابلیت توسعه پذیری و قابلیت دسترسی بالا، افزایش هزینه های عملیاتی با افزایش داده ها و ظرفیت محدود را نام برد.

۱-آشنایی با NoSQL

پایگاه داده های رابطه ای به طور گسترده در بسیاری از برنامه های کاربردی ، برای ذخیره و بازیابی داده ها استفاده می شوند. بیشترین کارایی پایگاه داده های رابطه ای زمانی است که از آن ها جهت رسیدگی به مجموعه ی محدودی از داده ها استفاده می شود. مدیریت حجم عظیمی از داده های بلادرنگ توسط پایگاه داده های رابطه ای ناکارآمد است . برای غلبه بر این مشکلات و رفع محدودیت های پایگاه داده های رابطه ای ، پایگاه داده های غیررابطه ای NoSQL بوجود آمدند (Carlo Strozzi., ۲۰۱۲). نخستین بار در سال ۱۹۹۸ عبارت NoSQL را برای اشاره به پایگاه های داده ی سبک و منبع باز رابطه ای به کار گرفت (که از رابط SQL استفاده نمی کردند). هرچند بعدها وی به این نکته اشاره کرد که این عبارت و مفهوم آن ، کاملًا از مدل رابطه ای جدا شده و دیگر بهتر است آن را NoRELبنامیم (George, ۲۰۱۳).

۱-۱- مزایا و معایب NOSQL

مزایای پایگاه داده Nosql شامل موارد زیر است :

خواندن و نوشتن سریع داده ها

حمایت از ذخیره سازی انبوه

توسعه ی آسان

هزینه ی کم

با این حال NoSql مانند هر پایگاه داده ی دیگری ، این پایگاه داده نیز دارای معایبی شامل موارد زیر است :

عدم پشتیبانی از SQL

فاقد تراکنش ها و گزارش ها

عدم توسعه یافتن به اندازه ی کافی ، جهت پشتیبانی از بسیاری از محصولات تولید شده در سال های اخیر (Han et al,2011).

۱-۲- طبقه بندی NOSQL

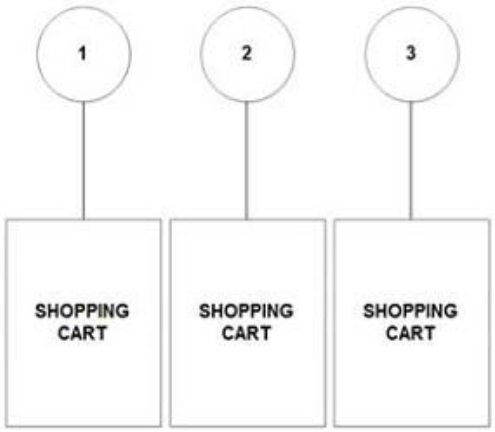
پایگاه داده های غیررابطه ای NoSql را می توان در چهار طبقه ی مختلف ، به شرح زیر دسته بندی کرد (Taura et al., ۲۰۱۲):

ISSN: 2745-3673
http://www.Rcsj.ir

ذخیره سازی مقدار-کلید
جدول بزرگ
پایگاه داده ی اسناد
پایگاه داده ی گراف

۱-۲-۱- ذخیره سازی مقدار-کلید

مدل داده ی مقدار-کلید به این مفهوم است که یک مقدار، مربوط به یک کلید است . اگرچه ساختار آن ساده تر از پایگاه داده ی رابطه ای است ، ولی سرعت پرس وجوی بالاتری دارد و از ذخیره سازی انبوه و همزمانی بالا پشتیبانی می کند. پرس وجوها و عملیات داده ای را از طریق کلید اصلی فراهم می کند. برای مثال ؛ در مورد یک سبد خرید نشان داده شده در شکل (۳)، هر سبد خرید در محفظه ی منحصر به فردی نشان داده شده است و با استفاده از یک مقدار-کلید که می تواند شناسه ی کاربر باشد، به آن اشاره می - شود (Taura et al, ۲۰۱۲):



شکل (۳) مثال سبد خرید برای ذخیره سازی مقدار- کلید (Taura et al., ۲۰۱۲)

۱-۲-۲- جدول بزرگ

موتور جستجوی Zvents، سیستم متن باز ذخیره سازی توزیع شده ی داده را با ترسیم جدول بزرگ توسعه داد. جدول بـزرگ یـک نقشه ی ذخیره شده ی چند بعدی ثابت ، پراکنده و واضح است . اندیس گذاری نقشه بوسیله ی یک کلید ردیف ، کلیـد سـتون ، و یـک برچسب زمان انجام می گیرد. در جدول بزرگ آرایه هایی از بایت های تفسیر نشده به عنوان مقادیر استفاده می شوند. جـدول بـزرگ داده های ساختاریافته را ذخیره می کند. هر نوع داده (متن تا اشیاء سریالیزه شده) می تواننـد در آن ذخیـره شـوند. ایـن مـدل داده محدودیت اندازه را برای هیچ مقداری تحمیل نمی کند و یک جدول می تواند تعداد نامحدودی ستون داشـته باشـد (Taura et al,2012) :

۱-۲-۳-پایگاه داده ی اسناد

پایگاه داده ی اسناد، همزمانی عملیات خواندن و نوشتن با کارایی بالا را برآورده نمی سازد، اما در عوض ذخیره سازی داده های بزرگ و کارایی پرس وجوها را به خوبی تضمین می کند (Taura et al., ۲۰۱۲).

۱-۲-۴- پایگاه داده ی گراف

این مدل داده با سه انتزاع اصلی کار می کند (Taura et al., ۲۰۱۲):
گره
ارتباط بین گره ها
جفت مقدار-کلید (که می تواند به گره ها و روابط وصل شود).

ISSN: 2745-3673
http://www.Rcsj.ir

پایگاه داده گراف هنگامی که داده ها را می توان به عنوان نمودار ارائه داد، مورد استفاده قرار می گیرد، به عنوان مثال در شبکه - هایاجتماعی (Abramova and Bernardino, ۲۰۱۳)

۲-معرفی کاساندرا

کاساندرا در آغاز توسط یک شبکه اجتماعی معروف برای پاسخ‌گویی به نیازهای روزافزون آن شبکه درتعاملات داده‌ای بسیار بزرگ ساخته شد و بعدها، توسعه و نگه‌داری آن به آپاچی محول‌شد. کاساندرا یک پایگاه‌داده اپن‌سورس است که براساس طرح Amazon Dynamoو مدل داده‌ای Bigtable (که از طرف گوگل ارائه شده) طراحی و توسعه داده‌شده است و قابلیت‌های کلیدی مانند توزیع یافتگی، تمرکز زدایی، مقیاس‌پذیری، دسترس‌پذیری بالا، مقاومت در مقابل خطا، ثبات تنظیم‌پذیر و ستون‌گرایی مدل داده‌ای در توسعه آن همواره مورد توجه بوده است.

به یقین، خلاصه‌کردن تمام قابلیت‌های کاساندرا در چند کلمه ساده، گویای واقعیت‌ها و ایده‌های موجود در پس توسعه چنین پایگاه داده‌ای نیست، اما سعی شده تا در مقاله‌های قبلی به گوشه‌ای از دلایل و مزایای این معماری جدید برای پایگاه‌های داده اشاره شود. (Lakshman and Malik, ۲۰۱۰).

هدف اصلی کاساندرا، قابلیت دسترس پذیری بالا، بدون داشتن هیچ نقطه ی شکستی است . علاوه بر این ، ویژگی های اصلی کاساندرا، پارتیشن بندی انعطاف پذیر، تکرار، عضویت و تشخیص شکست می باشد (Swamy, ۲۰۱۳). در حال حاضر کاساندرا به عنوان سیستم ذخیره سازی داخلی برای سرویس های مختلف در فیس بوک در حال توسعه است(Lakshman and Malik ,2010)

۲-۱- مدل داده ای کاساندرا

تا چند سال پیش، تقریباً تمام توسعه‌دهندگان برنامه‌هایی که به گونه‌ای با داده‌ها و پایگاه‌های داده سروکار داشتند، تنها مدل داده‌ای که برای طراحی در اختیار داشتند، مدل سنتی جدول، ستون و سطری بود که میراث حکمرانی چندین و چند ساله مدل پایگاه‌های داده رابطه‌ای‌بوده است. هم‌اکنون نیز بسیاری می‌پندارند همچنان تنها مدل قابل استفاده برای پیاده‌سازی پایگاه داده، همین مدل سنتی داده‌ای است. اما دنیا دیگر تغییر کرده و براساس چالش‌های موجود بر سر راه مدل سنتی، مدل‌های جدیدی معرفی شده‌اند که در بسیاری از زمینه‌ها و در مواجهه با بسیاری از چالش‌ها، شایستگی‌های بسیاری دارند. برای آموختن کاساندرا، بهتر است برای چند لحظه، تمام آنچه را که درباره پایگاه‌های داده می‌دانید، به فراموشی بسپارید. بسیاری از مفاهیم موجود در مدل داده کاساندرا مانند فضای نام یاkeyspace ، برای توسعه‌دهندگان قدیمی پایگاه داده، مفاهیمی جدید و ناآشنا بوده و بسیاری دیگر، مانند ستون یا column معانی متفاوتی با مشابه‌های قدیمی خود دارند. همچنین، با این‌که کاساندرا براساس مفاهیم بنیادی داینامو وBigtable ساخته شده است، اما مدل داده‌ای و مفاهیم مرتبط منحصر به فردی برای خود دارد.

۲-۲- معماری کاساندرا

با توجه به عملکرد پایگاه داده ی کاساندرا، معماری آن ، ویژگی های لازم مانند مقیاس پذیری ، قابلیت دسترسی بالا، تحمل پذیری خطا، همزمانی و مقاوم بودن را پشتیبانی می کند. کاساندرا مبتنی بر ترکیبی از مفاهیم Big Table گوگل و Dynamo آمازون تشکیل شده است . Big Table از سیستم فایل توزیع شده ای به نام سیستم فایل گوگل استفاده می کند که دارای تنها یک سرور اصلی است (که فراداده ها را برای داده های واقعی که در سرورهای مقیم ذخیره شده اند، نگهداری می کند). در حالی که دینامو مبتنی بر سیستم ذخیره سازی توزیع شده و با سازگاری نهایی است و سرویس هایی با قابلیت دسترسی بالا را فراهم می کند. در ادامه ویژگی های اساسی کاساندرا، بیان خواهند شد (Swamy, ۲۰۱۳).

۲-۳- پارتیشن بندی

کاساندرا ویژگی پارتیشن بندی را برای مدیریت حجم زیادی داده (که به پایگاه داده وارد شده اند) را پشتیبانی می کند.کاساندرا لیستی از گره ها را نگهداری می کند و داده ی ورودی را در میان گره ها به یکی از روشهای سازگار یا ناسازگار توزیع می کند. در پارتیشن بندی با استفاده از هماهنگ کننده ، از مسیریابی همه ی درخواست ها به یک گره ی خاص ، جلوگیری می شود (Swamy,2013).

ISSN: 2745-3673
http://www.Rcsj.ir

۲-۴- تکرار

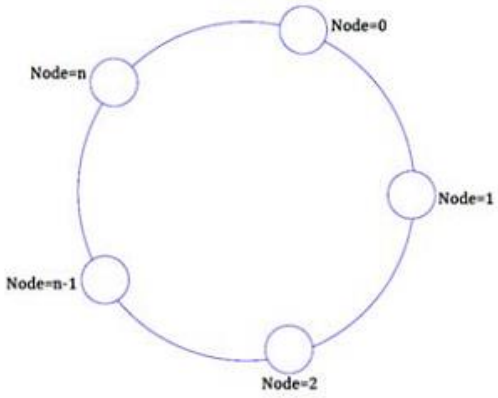
کاساندرا برای رسیدن به قابلیت دسترسی و ماندگاری بالا از تکرار استفاده می کند. هر قلم داده در Nمیزبان تکرار شده است . Nفاکتور تکرار پیکربندی شده در هر نمونه از کاساندرا است . هر زمان که یک گره نوشته می شود و یا بروزرسانی می شود، هماهنگ کننده بروز رسانی های مربوطه را بین همه ی ۱-Nگره ی باقی مانده پخش می کند. کاساندرا گزینه هایی را برای سیاست های تکرار مختلف در اختیار سرویس گیرنده ها قرار می دهد که شامل موارد زیر هستند:

استراتژی رک ناآگاه ۱۶: کپی ها همیشه به گره ی بعدی در طول حلقه فرستاده می شوند.

استراتژی رک آگاه ۱۷: یک کپی به اولین گره ی حلقه که متعلق به مرکز داده ی دیگر است ، فرستاده می شود و کپی های باقی مانده در صورت وجود، به اولین گره در طول همان حلقه فرستاده می شوند (Swamy, ۲۰۱۳).

۲-۵- عضویت

کاساندرا مکانیزم جدیدی از خود برای حفظ عضویت در میان گره های سیستم معرفی نمی کند. در واقع این پایگاه داده از پروتکل Scuttlebutt gossipبرای حفظ عضویت در میان گره های شرکت کننده در سیستم ، استفاده می کند. این پروتکل مسئول انجام بررسی های مکرر برای اطلاعات عضویت و پخش اطلاعات مربوطه ، به همه ی گره های دیگر است ؛ به طوری که تمام گره های مرزی سیستم ، اطلاعات مربوط به گره های موجود در سیستم را می دانند. مطابق شکل (۴) نمودار عضویت برای Nگره به روش دایره ای مرتب شده است . تابع هش ، خروجی (Key) % (N) =valueرا محاسبه می کند و داده به گره با شماره ی ترتیبی برابر با مقدار valueنسبت داده می شود (Swamy, ۲۰۱۳).



شکل (۴) نمودار عضویت (Swamy, ۲۰۱۳).

۲-۶- تشخیص شکست

تشخیص شکست مکانیزمی است که توسط آن یک گره می تواند به طور محلی تعیین کندکه آیا هر گره ی دیگر در سیستم هم اکنون فعال ۱۸ و یا غیرفعال ۱۹ است . برخلاف مکانیزم تشخیص خرابی قطعی سنتی ۲۰، کاساندرا با استفاده از یک مدل احتمالی کارامد بررسی می کند که یک گره فعال یا غیرفعال است . این روش مبتنی بر الگوریتم تشخیص خرابی تعهدی ۲۱ است که از پارامتر آستانه ی مشخص که با ϕنشان داده می شود، استفاده می کند. ارزش این پارامتر بر اساس شبکه ی محلی و وجود بار گره ی خاصی که قصد تشخیص خطای آن را داریم ، تنظیم می شود (Swamy,۲۰۱۳).

به عنوان بخشی از مکانیزم تشخیص شکست ، ماژول تشخیص خرابی تعهدی ، مقداری را منتشر می کند که به عنوان مقدار سطح تردید شناخته می شود و این واقعیت که گره جاری فعال یا غیر فعال است را منعکس می کند. اکنون احتمال اینکه این اظهارها نادرست باشد، بستگی به مقدار تنظیم شده برای آستانهϕ دارد. مقدار بیشتر آستانه ϕنشان دهنده ی این موضوع است که گره ی مظنون ، به طور واقعی با احتمال بیشتری غیر فعال است (Swamy, ۲۰۱۳).

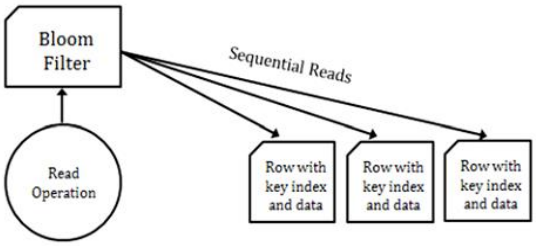
۲-۷- عمل Write

کاساندرا داده ها را در یک سیستم فایل محلی ماندگار می نویسد. داده ها به فرمتی ذخیره می شوند که خواندن کارآمد و قابل اعتماد را فراهم می کند. اگرچه به منظور بازیابی داده ها از خرابی احتمالی در سیستم قبل از ذخیره سازی در سیستم فایل ، آن ها را در

ISSN: 2745-3673
http://www.Rcsj.ir

CommitLogمی نویسد؛ بعد از تأیید سیستم ، داده های مربوطه که با موفقیت در CommitLogنوشته شده اند، به طور واقعی درون سیستم فایل مقیم بر روی دیسک سیستم نوشته می شوند. داده ها براساس شاخص های کلید تعریف شده برای هر ردیف به منظور جستجوی مؤثر و پردازش پرس وجوها ذخیره می شوند. هر عمل نوشتن به صورت یک فایل جداگانه نگهداری می شود. غالباً فرایند ادغام اجرا می شود و این فایل ها را در یک فایل بزرگ یکپارچه می کند.

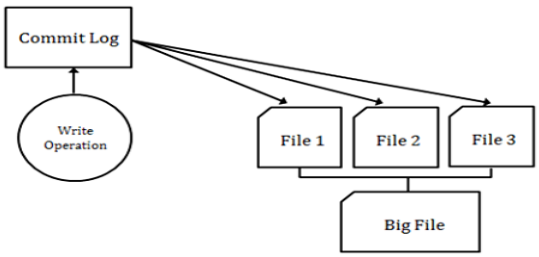
در شکل (۵) چگونگی عملیات نوشتن نشان داده شده است (Swamy, ۲۰۱۳).



شکل (۵) عمل نوشتن در کاساندرا (Swamy, ۲۰۱۳).

۸-۲- عمل Read

عملیات خواندن وظیفه دارد ساختار داده ی موجود در حافظه را به کمک شاخص های کلید تخصیص داده شده به ردیف ها بخواند. هر ردیف به طور خاص شناسایی شده و توسط کلید ستون تخصیص داده شده به آن ، اندیس گذاری می شود. در حالی که در جستجوی کلید ممکن است به محلی که حاوی کلید مربوطه نیست رجوع کرد. این سربار را می توان با فیلتر Bloomبرطرف کرد. این مکانیزم با جستجو در فایل هایی که به طور خلاصه شامل کلیدها هستند، از جستجو در فایل داده ای که شامل کلیدها نیست ، جلوگیری می کند. کاساندرا امکان جفت ستون (کلید و برچسب زمان) را ارائه می دهد که بر اساس ترتیب نزولی برچسب زمان مرتب شده اند، به طوری که آخرین رکورد همیشه در ابتدا دیده می شود. این امر مطابق با این واقعیت است که بسیاری از کاربران به آخرین نسخه ی دردسترس از داده ها نیاز دارند. در شکل (۶) چگونگی عملیات خواندن نشان داده شده است (Swamy , ۲۰۱۳).



شکل (۶) عمل خواندن در کاساندار (Swamy, ۲۰۱۳).

۳-زمینه های استفاده از کاساندرا

با وجود طراحی پیچیده و ویژگی های هوشمند کاساندرا، آن ابزار مناسبی برای هر زمینه کاری نیست . کاساندرا برای پروژه هایی با زمینه های کاری زیر مناسب است (Hewitt, ۲۰۱۰):

۱-۳- توسعه های بزرگ

در بسیاری از مهندسی های دقیق به دلیل ویژگی های دسترسی سریع ، سازگاری تنظیم پذیر و پروتکل نظیر به نظیر و قابلیت توسعه پذیری یکپارچه که از نقاط اصلی فروش کاساندرا محسوب می شوند، از آن استفاده شده است . هیچ کدام از این ویژگی های کیفیتی برای توسعه ی یک گره ، به تنهایی معنادار نیست . با این حال شرایط مختلف زیادی وجود دارند که در آن ها ممکن است به یک پایگاه داده ی رابطه ای تک گره نیاز داشته باشیم . برای پی بردن به اینکه از کدام نوع پایگاه داده استفاده کنیم ، نیاز به انجام برخی اندازه گیری ها و در نظر گرفتن ترافیک مورد انتظار و نیازهای عملیاتی داریم . اگر برآورد شود که می توان ترافیک را با یک سطح قابل قبول از کارایی با

ISSN: 2745-3673
http://www.Rcsj.ir

استفاده از تنها یک پایگاه داده ی رابطه ای خدمت رسانی کرد، استفاده از پایگاه داده ی رابطه ای ممکن است انتخاب بهتری برای این کار باشد، زیرا سیستم مدیریت پایگاه داده رابطه ای ۲۳ براحتی بر روی یک ماشین اجرا می شود و برای استفاده کنندگان آشناتر هستند. اگر حداقل نیاز به چندین گره برای پشتیبانی از یک کار باشد، در این صورت کاساندرا ممکن است انتخاب مناسبی باشد (Hewitt, ۲۰۱۰).

۴- یک نمونه پیاده سازی عملی از کاساندرا

در ادامه نحوه ی پیاده سازی یک پایگاه داده ی کاساندرا بیان خواهد شد.

۴-۱- پیاده سازی مدل داده

در کاساندرا برخلاف مدل رابطه ای که در آن کار طراحی از دامنه ی مفاهیم شروع می شود، کار با مدل داده ها شروع نمی شود، بلکه کار با مدل پرس وجو شروع به کار می شود. به طور نمونه برای طراحی یک "سیستم معاملات آنلاین "۲۴ براساس مدل رابطه ای چهار نقش در سیستم وجود دارند، که به همراه خصوصیاتشان در زیر لیست شده اند:

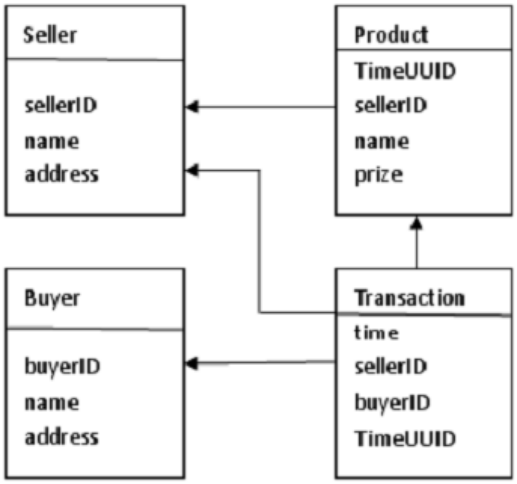
خریدار: شماره ی خریدار، نام خریدار، آدرس خریدار.

فروشنده : شماره ی فروشنده ، نام فروشنده ، آدرس فروشنده .

محصول : کد محصول ، شماره ی فروشنده ، نام محصول ، قیمت

تراکنش : زمان ، کد محصول ، شماره ی فروشنده ، شماره ی خریدار

شکل (۷) طراحی سیستم معاملات آنلاین بر اساس مدل رابطه ای (Wang and Tang, ۲۰۱۲)



شکل (۷) طراحی سیستم معاملات آنلاین بر اساس مدل رابطه ای (Wang and Tang, ۲۰۱۲)

در طراحی رابطه ای به ازای هر نقش یک جدول در نظر گرفته می شود. جدول تراکنش یک جدول اتصال است ، که برای نمایش یک رابطه ی چند به چند استفاده می شود. به منظور ساخت یک سیستم مبتنی بر کاساندرا، نیاز است که در ابتدا، پرس وجوها را تعیین شوند، سپس باید داده های مبتنی بر پرس وجوها سازماندهی شوند ، پرس جوها در این نمونه عبارتند از:

پرس وجوی اطلاعات فروشنده ؛ نظیر نام و آدرس فروشنده .

پرس و جوی اطلاعات خریدار؛ نظیر نام و آدرس خریدار.

پرس وجوی فروشنده ب محصول و خصوصیات خود محصول .

پرس و جوی تراکنش که در یک دوره زمانی خاص اتفاق افتاده است ؛ نظیر اطلاعات خریدار، فروشنده و محصول خریداری شده مدل مبتنی بر کاساندرا در شکل (۸) نشان داده شده است .

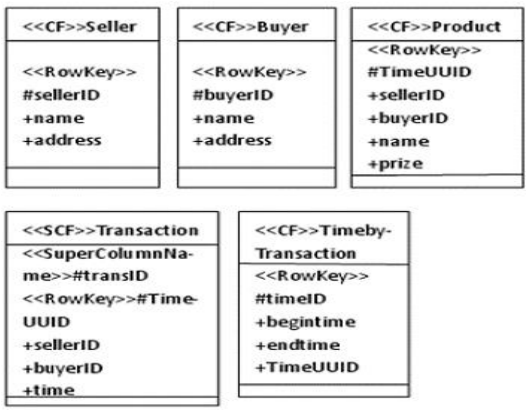
جداول فروشنده ، خریدار و محصول ، در مدل رابطه ای به طور مستقیم می توانند به خانواده ای از ستون ها در مدل کاساندرا تبدیل شوند. جدول تراکنش باید درون خانواده ی سوپرستون غیرنرمال سازی ۲۵ شود، زیرا زبان SQLدر کاساندرا وجود ندارد و باید یک شاخص در فرم خانواده ی ستون TimebyTransactionایجاد کرد.

ISSN: 2745-3673
http://www.Rcsj.ir

جهت پیاده سازی سیستم معاملات آنلاین براساس مدل کاساندرا ، یکی از خانواده ی ستون ها سوپر ستون های استفاده شده در ادامه شرح داده خواهد شد. به عنوان نمونه ؛ خانواده ی ستون فروشنده انتخاب می شود. در خانواده ی ستون فروشنده ، شناسه ی فروشنده (sellerID) به عنوان کلید خانواده ی ستون استفاده شده است . هر خصوصیت از فروشنده ، می تواند توسط یک ستون ذخیره شود و نام ستون همان نام خصوصیت است . ساختار ستون به صورت زیر است :

<ColumnFamily Name=”Seller”
CompareWith= ”UTF8Type”.>

داده ها به فرم زیر ذخیره می شوند:



شکل (۸) طراحی سیستم معاملات آنلاین بر اساس مدل کاساندرا (Wang and Tang, ۲۰۱۲)

Seller={ ..ColumnFamily
Bill021:{ ..ColumnFamily Key
{name:”name”, value:”Bill Wang”, timestamp:1234567},
{name:”address” ,value: ” Shanghai” timestamp:1234567}},
..first Seller
Lily028:{ ..CoumnFamily Key
{name:”name”, value:”Lily Qing”, timestamp:1234567},
{name:”address”, value:”Chengdu”, timestamp:1234567},
}, ..second Seller ... }

برای خانواده ابرستون ، به عنوان نمونه جدول تراکنش انتخاب شده است . شناسه ی تراکنش (transID) به عنوان کلید در نظر گرفته شده است و TimeUUIDنام سوپرستون است . همه ی تراکنش ها می توانند براساس ساعات تجارت مرتب شوند. بنابراین TimeUUIDTypeنقش مرتب کردن را بر عهده دارد. ساختار سوپر ستون به صورت زیر است (Wang and Tang, ۲۰۱۲, ۱۳۳۳):

<ColumnFamily Name=”Transaction”
ColumnType=”Super”
CompareWith=”TimeUUIDType”
CompareSubcolumnsWith=”UTF8Type” .>
The data stored in the form of:
Transaction={ ..Super ColumnFamily
12022011:{ ..firsttimeID, Key
{name:”1234rt0-b3j6-j6k8gjk975”,
..firstTimeUUID
value:{{name:”sellerID”,value:”Bill021”,timestamp:123456},
{name:”buyerID”,value:”Hill007”,timestamp:123456},{name:”t

ISSN: 2745-3673
http://www.Rcsj.ir

ime",value:"2011.11,12:03",timestamp:123456}},
{name:"1234rt0-b3j6-j35k5f7j788", ..secondTimeUUID
value:{...}
}, ...},
13022011:{ ..secondtimeID, Key
{name:...
value:…}, … } … }

ISSN: 2745-3673
http://www.Rcsj.ir

۵- نتیجه‌گیری

هدف از این مقاله بیان محدودیت های پایگاه داده های سنتی و نیز بیان مزایا و معایب استفاده از پایگاه داده های NoSql و مقایسه ی آن با پایگاه داده ی سنتی است . کاساندرا یک نمونه از پایگاه داده های غیررابطه ای است که در این مقاله بصورت کامل شرح داده شد. اگرچه بزرگترین نصب و راه اندازی کاساندرا در فیس بوک بوده است ، اما در حال حاضر بسیاری از شرکت ها کاساندرا را برای استفاده در پروژه های مختلف تولید، مورد ارزیابی قرار داده اند. در حال حاضر، کاساندرا توسط cisco و G4 platform شروع به استفاده شده است . همچنین Comcast و bee.tv برای پخش تلویزیون شخصی بر روی وب و تلفن همراه آن را مورد استفاده قرار داده اند. شرکت Riptano، توسعه نسخه جدیدکاساندرا را از سال ۲۰۱۰ آغاز کرده است و در حال افزودن ویژگی های بیشتر و گزینه های پشتیبانی بهتری برای آن است . با توجه به روند رو به رشد فناوری ، کاساندرا همواره نیاز به توسعه بیشتر دارد. زمینه های تحقیقاتی آینده کاساندرا، شامل افزودن فشرده سازهای کارا، توانایی پشتیبانی از کلیدهای سراسری اتمی ، پشتیبانی از اندیس ثانویه و مکانیزم های رمزنگاری خودکار داده ها می باشند.

ISSN: 2745-3673
http://www.Rcsj.ir

منابع و مراجع

[1] http://cassandra.apache.org/download/
[2] http://wiki.apache.org/thrift/ThriftInstallationWin32
[3] Abramova,V. and Bernardino, J. (2013); "NoSQL databases: MongoDBvs Cassandra", Presented at Proceedings of
[4] the International C* Conference on Computer Science and Software Engineering:14-22.
[5] Han, J., Haihong, E., Le, G. and Du, J. (2011); "Survey on NoSQL database", Presented at 6th international
[6] conference on Pervasive computing and applications, Port Elizabeth: 363-366.
[7] Hewitt, E. (2010); Cassandra: the definitive guide: O'Reilly Media.
[8] Lakshman, A. and Malik, P. (2010); "Cassandra: a decentralized structured storage system", ACM SIGOPS
[9] Operating Systems Review, vol. 44, no. 2: 35-40.
[10] Swamy, R. (2013); "Explicit Data Encryption Architecture for Cassandra", International Journal of Engineering
[11] Research& Innovation, vol. 2, no. 4: 376-379.