



عصری که پایگاه‌های داده NoSQL، پایگاه‌های داده توزیع شده، انبار داده و پایگاه‌های داده مبتنی بر شتاب‌دهنده-پردازنده گرافیکی محبوب و همه‌گیر شده‌اند، فراموش کردن تدریجی پایگاه‌های داده رابطه‌ای سنتی که همچنان به ذخیره و میزبانی اطلاعات حیاتی می‌پردازند چندان عجیب و غیر منتظره نیست. مبادا پیدایش پایگاه‌های داده‌ای همچون PostgreSQL و Oracle Database، Microsoft SQL Server MySQL/MariaDB به دهه 80 باز می‌گردد، با این وجود پایگاه‌های داده‌ای فوق همچنان فعال و در حال توسعه هستند و این فرآیند تنها به برطرف کردن باگ‌ها و تغییرات کوچک در وضعیت عملکرد خلاصه نمی‌شود. در این مقاله به معرفی ترفندهایی خواهیم پرداخت که عملکرد پایگاه‌های داده SQL را بهبود می‌بخشند.

1- جست‌وجوی متن کامل

شاخص‌های (indexes) پایگاه‌های داده رابطه‌ای استاندارد برای بهبود عملکرد در اغلب موارد از فیلدهای کوتاه یا حتی هاش استفاده کنند، با این وجود، جست‌وجوی متن کامل (Full-text search) موضوع دیگری است. جست‌وجوی متن کامل توسط نوع دیگری از شاخص‌ها انجام می‌شود که بیشتر ماهیتی شبیه به یک فهرست معکوس از کلمه‌ها، رکوردها و موقعیت‌های متناظر به آن‌ها دارند. در یک جست‌وجوی متن کامل فهرستی از کلمه‌های توقف وجود دارد که البته نادیده گرفته می‌شوند و از یک الگوریتم استفاده می‌شود. پایگاه‌های داده Oracle Database، Microsoft SQL Server، MySQL/MariaDB و PostgreSQL همگی امکان استفاده از جست‌وجوی متن کامل روی انواع خاصی از فیلدهای متنی همچون CHAR، VARCHAR و TEXT که به شاخص‌های متن کامل مجهز هستند را فراهم می‌کنند. شما می‌توانید از یک موتور جست‌وجوی متن کامل همچون Elasticsearch یا Solr برای شاخص‌گذاری و جست‌وجوی متن در یک پایگاه داده استفاده کنید.

2- داده‌های JSON

نشانه‌گذاری شی جاوااسکریپت (JSON) سرنام JavaScript Object Notation با فراگیر شدن جاوا اسکریپت در وب اهمیت پیدا کرد و به یکی از فرمت‌های داده استاندارد در پایگاه‌های داده اسناد NoSQL تبدیل شد. همچنین، خیلی از پایگاه‌های داده مبتنی بر SQL نیز پشتیبانی از JSON را به امکانات خود اضافه کردند. البته هر پایگاه داده‌ای روش اختصاصی خود برای پشتیبانی از JSON ارائه کرده است. به عنوان مثال، در SQL Server شما می‌توانید به شکل مستقیم روی فیلدهای متنی، پرس‌وجویی برای دریافت مقادیر JSON انجام داده یا اسناد JSON را به جدول تبدیل کنید. همچنین، امکان اعمال محدودیت برای فیلدهای متنی برای اطمینان از این موضوع که آن‌ها به درستی با فرمت JSON ایجاد می‌شوند، وجود دارد. از طرف دیگر PostgreSQL از یک نوع مشخص از JSON و توابع مرتبط با آن استفاده می‌کند.

3- داده‌های XML

XML نوعی زبان نشانه‌گذاری گسترش‌پذیر است که فرمت کلی نشانه‌گذاری متن‌ها را تعیین می‌کند و همچون JSON از یک استاندارد داده نیمه ساختار یافته استفاده می‌کند. در روزهایی که خدمات وب و فراخوانی‌های بازگشتی وب از طریق AJAX اختراع شدند، این نوع فرمت تبادل اطلاعات رواج پیدا کرد و از آن زمان در بسیاری از برنامه‌ها جایگزین JSON شد. XML مجموعه‌ای از فناوری‌های دیگر را درون خود جای داده است. به‌طور مثال، XPath زبانی برای آدرس‌دهی بخش‌های متفاوت یک سند XML است و از XSLT برای تبدیل ساختار یک سند XML، مثل تولید صفحات وب از داده‌های XML استفاده می‌کند. همه پایگاه‌های داده رابطه‌ای یاد شده به شکلی از داده‌های XML پشتیبانی می‌کنند. استفاده از قابلیت ذخیره‌سازی XML در جداول رابطه‌ای می‌تواند برای برخی برنامه‌ها مفید باشد.

مطلب پیشنهادی



نگاهی به مراحل پیش از تولد، رشد، جوانی، میان‌سالی، پیری و مرگ مراکز داده
داستان زندگی یک مرکز داده؛ ساخت، نگهداری و پایان کار

4- داده‌های جغرافیایی

به نظر می‌رسد اطلاعات جغرافیایی و ژئومتریک را باید به شکل عددی استفاده کرد، اما انواع محاوره‌هایی که اغلب روی داده‌های فضایی (spatial) اجرا می‌کنید به یک شاخص نیاز دارند که داده‌های دو بعدی را تشخیص می‌دهد. به‌عنوان مثال، ممکن است بخواهید فاصله بین دو فرودگاه، نزدیک‌ترین مرکز خرید یا هتل به یک مکان خاص را بدانید. برای این نوع از محاوره‌ها باید از یک شاخص R-tree، SP-GiST، quadtree یا UB-tree استفاده کنید. تمام پایگاه‌های داده رابطه‌ای یاد شده از اطلاعات و شاخص‌های فضایی پشتیبانی می‌کنند، هر چند برخی از آن‌ها به یک افزونه همچون PostGIS برای PostgreSQL احتیاج دارند. آن‌ها به‌طور معمول هماهنگ با ویژگی Open Geospatial Consortium Simple Features و استاندارد SQL/MM Spatial ISO/IEC هستند.

5- جداول درون حافظه

به استثنای PostgreSQL که فرآیند کش کردن درون حافظه را آن هم نه به‌طور کامل برای جداول درون حافظه انجام می‌دهد، تمام پایگاه‌های داده رابطه‌ای که اشاره شد این پتانسیل را دارند تا جداول را درون حافظه ایجاد کنند. جداول درون حافظه به میزان قابل ملاحظه‌ای سرعت کار با اطلاعات درون این نوع جداول را به ویژه در شرایطی که حجم زیادی از داده‌ها باید خوانده شوند، افزایش می‌دهند. آمارها نشان می‌دهند گاهی اوقات این افزایش سرعت تا 30 برابر هم رسیده است.

اغلب محدودیت‌هایی برای جداول درون حافظه وجود دارد. اولین و اصلی‌ترین محدودیت به میزان حافظه اصلی باز می‌گردد که جداول را درون خود نگهداری می‌کند. دومین محدودیت در ارتباط با پایگاه‌های داده و موتورهای خاص است. به‌طور مثال، موتور ذخیره‌سازی MySQL MEMORY ناپایدار و موقتی است (وقتی سرور MySQL با خطا مواجه شده یا متوقف می‌شود این حافظه ناپدید می‌شود) یا اگر تراکنش‌ها، کلیدهای خارجی، انواع داده‌های جغرافیایی و شاخص جست‌وجوی متن کامل به نوشتن حجم انبوهی از داده‌ها نیازمند باشند به درستی کار نخواهند کرد. موتور MySQL NDB Cluster توانسته برخی از این محدودیت‌ها را برطرف کند، اما برای اجرا به نسخه خاصی از MySQL احتیاج دارد و باید چند نوع گره فعال در خوشه داشته باشد.

SQL Server In-Memory OLTP از جداول درون حافظه بهینه شده استفاده می‌کند که همزمان می‌تواند از جداول پایدار و تراکنش‌ها و همین‌طور جداول ناپایدار پشتیبانی کند.

Oracle Database In-Memory به خوبی عملکرد پایگاه داده را برای تحلیل‌های بلادرنگ (OLAP) و بار کاری ترکیبی (HTAP) بهینه‌سازی می‌کند.

In-Memory Column Store یکی از قابلیت‌های کلیدی Oracle Database In-Memory است.

6- منابع داده خارجی

نسخه‌های جدید اس‌کیوال سرور از راهکاری قدرتمند برای پیاده‌سازی محاوره‌ها روی منابع داده خارج از جداول خود

پایگاه داده استفاده می‌کند. شما می‌توانید از Hadoop، blob storage یا یک پایگاه داده رابطه‌ای برای ساخت یک EXTERNAL DATA SOURCE استفاده کرده و در ادامه از PolyBase یا Azure Elastic Database queries (v12 SQL Database+) به منظور پیاده‌سازی محاوره‌ها یا بارگیری داده‌ها روی منابع داده خارجی استفاده کنید. PostgreSQL Foreign Data Wrappers به محاوره‌های PostgreSQL اجازه می‌دهد روی مجموعه گسترده‌ای از منابع داده راه دور اجرا شوند. منابع داده‌ای فوق، پایگاه‌های داده SQL، NoSQL و زیرساخت‌های بزرگ داده‌ها و حتی فایل‌ها را شامل می‌شوند. Foreign Data Wrappers از استاندارد SQL/MED سرنام SQL Management of External Data پیروی می‌کند.

7- خوشه‌های بزرگ داده‌ها

پیش‌نمایش SQL Server 2019 نشان داد، خوشه‌های بزرگ داده‌های مایکروسافت اس‌کیوال سرور این امکان را می‌دهند تا خوشه‌های گسترش‌پذیر اس‌کیوال سرور، اسپارک و کانتینرهای HDFS در حال اجرا در کوبرنتیس را پیاده‌سازی کنید. این مولفه‌ها در کنار یکدیگر اجرا می‌شوند تا امکان خواندن، نوشتن و پردازش بزرگ داده‌ها از Transact-SQL (از طریق PolyBase) یا اسپارک را فراهم کرده و اجازه می‌دهند داده‌های رابطه‌ای با ارزش بالای خودتان را با بزرگ داده‌های با حجم بالا ترکیب کرده و تحلیل کنید. اوراکل یک چنین راهکاری در محصولات Oracle Big Data Appliance On و Oracle Big Data Cloud Service Premises ارائه داده است.

8- Read replicas (خواندن کپی‌ها)

مای‌اس‌کیوال، MariaDB و PostgreSQL همگی قابلیت خواندن کپی را فراهم می‌کنند. این ویژگی توان عملیاتی خواندن مستقیم داده‌ها را افزایش داده، اما از طرف دیگر می‌تواند به‌طور غیر مستقیم عملکرد نوشتن را با کاهش زمان بارگیری در سرور افزایش دهد. MySQL/MariaDB از چند نوع نسخه کپی که شامل موارد زیر می‌شود، پشتیبانی می‌کنند. log-file and GTID-based; asynchronous, synchronous using in-memory NDB Clusters, semi-synchronous, and delayed; statement-based, row-based, mixed-base PostgreSQL متفاوتی برای این منظور ارائه کرده است. Amazon Aurora از برنامه خواندن نسخه کپی اختصاصی خود برای MySQL و PostgreSQL استفاده می‌کند. Aurora امکان دسترسی به بیش از 15 نسخه کپی را فراهم می‌کند که زمان تاخیر همگام‌سازی در آن‌ها کمتر از 20 میلی ثانیه است. SQL Server از یک روش خواندن نسخه کپی ثانویه در یک خوشه active-active پشتیبانی می‌کند. Oracle Database از محیط‌های یک طرفه فقط خواندنی نسخه کپی با استفاده از کپی فوری از جدول فقط خواندنی پشتیبانی می‌کند. Oracle Enterprise از قابلیت‌های پیشرفته نسخه کپی پشتیبانی می‌کند که با فراهم کردن امکان به‌روزرسانی جداول نسخه کپی توسط برنامه‌ها امکانات نسخه کپی فقط خواندنی را گسترش می‌دهد.

مطلب پیشنهادی



چگونه عمر یک مرکز داده را افزایش دهیم؟

9- Sharding

Sharding روشی برای تقسیم داده در سرورها است. تقسیم‌بندی عمودی شامل توزیع جداول مختلف در سرورهای مختلف است. به‌عنوان مثال، قرار دادن ماهیتی روی یک سرور، سفارش در دیگری و جمع‌آوری جدول‌ها روی سرور سوم برای تجزیه و تحلیل از جمله این موارد است. رویکرد فوق زمانی که قصد دارید جدول‌ها از چند سرور مختلف با یکدیگر ترکیب شوند، مشکلی به وجود می‌آورد. تقسیم‌بندی افقی نیز شامل تقسیم جداول مستقل روی سرورها است. چیزی شبیه به صندلی‌های ثابت شده در یک همایش که بر اساس حروف الفبا مرتب شده‌اند. انجام تقسیم‌بندی افقی به صورت دستی کاری طاقت‌فرسا است، اما خوشبختانه گزینه‌های مختلفی برای تقسیم‌بندی افقی خودکار وجود دارد.

Citus تقسیم‌بندی افقی خودکار را درون PostgreSQL انجام می‌دهد. همچنین، Citus یک تعدیل‌کننده گره را به عنوان front-end برای گره‌هایی که داده‌ها را نگهداری می‌کنند ارائه می‌کند. Microsoft Azure SQL Database v12 نیز از یک نقشه پایگاه داده انعطاف‌پذیر برای خودکارسازی تقسیم‌بندی افقی پشتیبانی می‌کند.

10- راهکارهای ذخیره‌سازی فراتر از SQL

هر پایگاه داده رابطه‌ای افزونه‌های SQL مخصوص به خود را برای رویه‌های ذخیره‌سازی در اختیار دارد که از آن جمله می‌توان به Transact-SQL برای SQL Server و PL/SQL برای Oracle اشاره کرد. طی سال‌های اخیر پایگاه‌های داده رابطه‌ای با زبان‌های برنامه‌نویسی و امکانات یادگیری ماشین یکپارچه شده‌اند. یکی از اولین نمونه‌ها در این مورد اجرای جاوا در Oracle Database به عنوان یک قابلیت الحاقی به PL/SQL است. از آن زمان ما شاهد افزونه‌های پایگاه داده برای زبان‌های Python, PHP, Perl, C#, C++، C و R بوده‌ایم. معرفی پایگاه‌های داده NoSQL در سال 2009 نیاز به پایگاه‌های داده با گسترش‌پذیری و سطح دسترسی بالا برای برنامه‌های وب را افزایش دادند. طی یکی دو سال اخیر، پایگاه‌های داده کلاسیک گزینه‌های زیادی برای بهبود گسترش‌پذیری و بهبود سطح دسترسی اضافه کرده‌اند. قبل از کنار گذاشتن پایگاه داده SQL قدیمی خود و مهاجرت به NoSQL، حتما اسناد مربوط به پایگاه داده قدیمی را بررسی کنید. بدون شک می‌توانید از هر دو اکوسیستم استفاده کرده و بهترین نتیجه را دریافت کنید.

تاریخ انتشار:

05 مهر 1398

نشانی منبع:

<https://www.shabakeh-mag.com/workshop/16062/%DB%B1%DB%B0-%D9%82%D8%A7%D8%A8%D9%84%DB%8C%D8%AA-%D9%BE%D9%86%D9%87%D8%A7%D9%86-%DA%A9%D9%87-%D9%BE%D8%A7%DB%8C%DA%AF%D8%A7%D9%87-%D8%AF%D8%A7%D8%AF%D9%87-%D9%82%D8%AF%DB%8C%D9%85%DB%8C%E2%80%8C-%D9%85%DB%8C%E2%80%8C%D8%AA%D9%88%D8%A7%D9%86%D8%AF-%D8%A7%D9%86%D8%AC%D8%A7%D9%85-%D8%AF%D9%87%D8%AF>