



مقایسه پایگاه‌های داده توزیع شده: Amazon و Google Spanner, Cockroach DB Aurora

محمدجواد حسین پور

عضو هیئت علمی بخش مهندسی کامپیوتر، دانشگاه آزاد اسلامی واحد استهبان، استهبان، ایران

فاطمه زارع

دانشجوی کارشناسی ارشد دانشگاه آزاد اسلامی واحد استهبان، استهبان، ایران

چکیده

در سال‌های اخیر، نیاز به پایگاه‌های داده توزیع شده برای پشتیبانی از مقیاس‌پذیری بالا و در دسترس بودن دائمی افزایش یافته است. Amazon Aurora، Google Spanner، Cockroach DB و Aurora به سیستم پیشرو در این حوزه هستند. این مقاله به مقایسه این سه سیستم از نظر معماری، سازگاری ACID، مقیاس‌پذیری، تاخیر و عملکرد می‌پردازد. نتایج نشان می‌دهد که هر سیستم برای موارد استفاده خاصی مناسب است؛ برای مثال، Spanner به دلیل سازگاری قوی با ACID برای سیستم‌های بانکی مناسب‌تر است، در حالی که Cockroach DB برای مقیاس‌پذیری افقی بهتر عمل می‌کند.

واژگان کلیدی: پایگاه داده، پایگاه داده توزیع شده، Amazon، Cockroach DB، Google Spanner، Aurora



مقدمه

«توزیع» (Distribution) یا ادغام قابلیت‌های پردازشی چندین ماشین برای استفاده از کارایی آن‌ها به گونه‌ای که گویی همگی یک موجودیت واحد هستند، در دنیای امروزی محاسبات، بسیار رایج و مرسوم است. به احتمال زیاد، بسیاری از افراد تاکنون بدون اینکه مطلع باشند، با یک سیستم توزیع شده تعامل داشته‌اند. برای مثال، اپلیکیشن‌های شبکه‌های اجتماعی به کاربران خود تجربه‌ای رفتاری را ارائه می‌دهند که بر قابلیت و عملکرد کامپیوتری قدرتمند، اما واحد دلالت دارد، در حالی که در فناوری زیربنایی آن اصلاً این چنین نیست. پلتفرم‌هایی مثل توئیتر و فیسبوک در مقیاس مناسب اجرا می‌شوند و بر بستر ابزارهایی ساخته شده‌اند که در میان تعداد زیادی از عناصر محاسباتی، اغلب در مکان‌های متفاوت به اشتراک‌گذاری (پراکنده) شده و در قالب برنامه‌های نرم افزاری منظم و مرتب و به راحتی قابل استفاده‌اند و به خوبی در هم ادغام شده‌اند. کاربران اغلب هیچ اطلاعی راجع به معماری زیربنایی محصولات نرم‌افزاری مورد استفاده خود ندارند. پایگاه داده‌ها در حوزه ساخت و توسعه اپلیکیشن نقش بسیار برجسته و مهمی دارند و همچنین به میزان قابل توجهی از قابلیت «توزیع» بهره می‌برند. پایگاه داده‌های توزیع شده، بر خلاف دیتابیس‌هایی که در یک ماشین واحد تعبیه شده‌اند، قابل افزایش مقیاس هستند، کارایی زیادی دارند و شفافیت داده‌ها نیز در آن‌ها ملموس است. هر پایگاه داده تحت شبکه‌ای را نمی‌توان پایگاه داده توزیع شده نامید. در واقع برای جای گرفتن یک دیتابیس در دسته پایگاه داده‌های توزیع شده، پیش‌شرط‌هایی شامل ذخیره‌سازی توزیع شده داده‌ها و اجرای پرسمان، امنیت، یک API پرسمان یکپارچه و بهینه‌سازی هزینه‌ها باید وجود داشته باشند.

به طور کلی پایگاه‌های داده توزیع شده اساساً مجموعه‌هایی از داده‌های منطقاً مرتبط و مصنوعات پایگاه داده‌ای هستند که در گره‌هایی از سیستم توزیع شده واقع شده‌اند که به این گره‌ها، «Site» (جایگاه | مقر | محل | سایت) هم می‌گویند. پایگاه داده‌های توزیع شده بر وابستگی منطقی و فیزیکی به داده‌ها موجود در مدل‌های محاسباتی اولیه چیره شده‌اند و از محدودیت‌های مقیاس‌بندی و پردازشی موجود در پایگاه داده‌های متمرکز محبوس در توان پردازشی تک‌گره‌ای عبور کرده‌اند.

پایگاه داده‌های توزیع شده دارای ۴ مزیت اصلی متمایز، شامل مقیاس‌بندی، عملکرد، قابلیت اطمینان و شفافیت هستند که این نقاط قوت باعث می‌شوند پایگاه داده‌های توزیع شده برای کاربردهای نیازمند داده‌های فراوان، از جمله تشخیص کلاهبرداری آنی، تجزیه و تحلیل، تشخیص چهره و سایر موارد ایده‌آل باشند. با دلایل ذکر شده، پایگاه‌های داده نقش اساسی در ذخیره و مدیریت اطلاعات دارند. با گسترش خدمات ابری و نیاز به داده‌های توزیع شده، استفاده از پایگاه‌های داده توزیع شده به یک ضرورت تبدیل شده است. این پایگاه‌ها به سازمان‌ها اجازه می‌دهند داده‌های خود را به صورت هم‌زمان در چندین مکان ذخیره کنند و در صورت بروز خطا، دسترسی به داده‌ها همچنان امکان‌پذیر باشد. پایگاه داده‌های توزیع شده زیادی معرفی شده‌اند که در این مقاله، سه پایگاه داده توزیع شده مشهور یعنی Cockroach DB، Google Spanner و Amazon Aurora مورد بررسی و مقایسه قرار می‌گیرند.

کاربرد های پایگاه داده توزیع شده

سیستم های پایگاه داده توزیع شده به طور رایج در فعالیتهایی کاربرد دارند که در آنها به میزان زیادی نیاز به کار با داده ها وجود دارد و از داده های حجیم استفاده می شود. از جمله چنین فعالیت ها و حوزه هایی می توان به سیستم های شناسایی کلاهبرداری (تشخیص ناهنجاری)، تجزیه و تحلیل لحظه ای داده ها و تشخیص چهره اشاره کرد.

– کاربرد پایگاه داده توزیع شده در تشخیص کلاهبرداری:

پایگاه داده های توزیع شده به عنوان مخازن داده در تشخیص کلاهبرداری استفاده می شوند که به لحاظ محاسباتی یکی از پیچیده ترین و سنگین ترین حوزه ها به حساب می آید. تشخیص کلاهبرداری معمولاً شامل استفاده از یادگیری ماشین برای شناسایی، نظارت و هماهنگی واکنش ها به رخداد تخلف های فریبنده می شود. چنین سیستم هایی از تحمل خطا در پایگاه داده های توزیع شده برای مدیریت دفتر حساب تهدیدهای شناسایی شده استفاده می کنند.

– کاربرد پایگاه داده توزیع شده در تجزیه و تحلیل بلادرنگ:

با تجزیه و تحلیل بلادرنگ در درجه اول به محض جمع آوری داده ها، دغدغه بدست آوردن بینش هایی را در آنها داریم. داشبوردهای بلادرنگ که به وسیله آنها مصورسازی داده های جمع آوری شده بلافاصله بعد از پردازش از چندین منبع مختلف ارائه می شود، بخش زیادی از سرعت شان را مدیون نزدیک بودن به هر یک از چندین نسخه برابر اصل در سیستمی توزیع شده هستند.

– کاربرد پایگاه داده توزیع شده در تشخیص چهره:

درست شبیه به تجزیه و تحلیل برخط، تشخیص چهره نیز در صورتی به بهترین نحو قابل پیاده سازی خواهد بود که از پایگاه داده توزیع شده با سرعت بالا و به منظور تضمین در دسترس بودن داده ها به وسیله کپی های نزدیک به محل استفاده شود. تشخیص چهره رویکردی در حوزه یادگیری ماشین به حساب می آید که در آن چهره افراد در محتواهای تصویری مثل عکس، تصویر لحظه ای از فریم های ویدیو یا سایر مواردی از این دست با تعداد زیادی از تصاویر در یک پایگاه داده بزرگ از تصاویر با هدف شناسایی اشخاص مطابقت داده می شود.

بررسی پایگاه های داده

– پایگاه داده Google Spanner

Google Spanner یک پایگاه داده توزیع شده مبتنی بر SQL است که ویژگی های سازگاری ACID، مقیاس پذیری افقی و در دسترس بودن بالا را به طور همزمان ارائه می دهد. این سیستم به دلیل توانایی اجرای تراکنش های توزیع شده هماهنگ شده در سطح جهانی شناخته شده است. یکی از برجسته ترین ویژگی های Spanner استفاده از فناوری TrueTime است که برای مدیریت هماهنگی سراسری بین سرورها استفاده می شود.

Google Spanner توسط Google Cloud به عنوان یک سرویس مدیریت شده پایگاه داده ارائه می شود و برای کاربردهای سازمانی که به قابلیت اطمینان بالا، سازگاری ACID، مقیاس پذیری جهانی و تاخیر پایین نیاز دارند، طراحی شده است. از جمله ویژگی های کلیدی Google Spanner میتوان به موارد زیر اشاره کرد.

- فناوری True Time: یک سیستم ساعت جهانی توزیع شده است که از طریق ترکیب ساعت های اتمی و ساعت های GPS برای هماهنگ کردن زمان بین مراکز داده استفاده می کند. این فناوری به Spanner امکان می دهد که تراکنش های توزیع شده را سازگار با ACID مدیریت کند و قفل گذاری زمانی (Timestamp Locking) را به صورت دقیق انجام دهد.

True Time امکان انجام تراکنش های چند منطقه ای (multi-region) را فراهم می کند و تضمین می کند که همه گره ها در سراسر جهان به یک زمان هماهنگ دسترسی دارند.

- پشتیبانی از تراکنش های ACID: برخلاف بسیاری از پایگاه های داده توزیع شده که فقط سازگاری نهایی (Eventual Consistency) را ارائه می دهند، Spanner سازگاری ACID قوی را فراهم می کند. این ویژگی برای سیستم های مالی، بانکی و مواردی که به داده های دقیق و تراکنش های اتمی نیاز دارند، بسیار مفید است.

- مقیاس پذیری افقی: می تواند به راحتی بین مراکز داده در مناطق جغرافیایی مختلف گسترش یابد و به دلیل معماری توزیع شده خود، مقیاس پذیری افقی دارد. هنگامی که حجم داده ها افزایش می یابد، گره های بیشتری اضافه می شوند و به این ترتیب سیستم بدون قطعی مقیاس پذیر می ماند. از آنجا که Google Spanner به صورت توزیع شده جهانی طراحی شده، این امکان وجود دارد که یک پایگاه داده واحد به صورت هم زمان در چندین منطقه جهان توزیع شود.

- SQL و پشتیبانی از مدل رابطه ای: برخلاف برخی پایگاه های داده NoSQL که از زبان های پرس و جوی سفارشی استفاده می کنند، Spanner از SQL استاندارد برای پرس و جوی داده استفاده می کند. این ویژگی به توسعه دهندگان اجازه می دهد از مهارت های SQL موجود خود برای کار با Spanner استفاده کنند. Google Spanner از جداول رابطه ای، کلیدهای اولیه، کلیدهای خارجی و محدودیت های یکپارچگی پشتیبانی می کند.

- دسترسی بالا و تحمل خطا: به دلیل طراحی توزیع شده، در دسترس بودن بالا (High Availability) را تضمین می کند. داده ها در چندین منطقه جغرافیایی تکرار می شوند، بنابراین حتی اگر یک مرکز داده از کار بیفتد، عملیات سیستم تحت تأثیر قرار نمی گیرد. این پایگاه داده از پروتکل Raft برای هماهنگ سازی نسخه های مختلف داده در سرورها استفاده می کند، که باعث می شود داده ها به صورت خودکار همگام شوند.

- مدیریت خودکار شاردینگ (Auto-sharding): به صورت خودکار داده ها را به Shard های کوچک تر تقسیم می کند و به هر گره در خوشه توزیع می کند. این ویژگی تضمین می کند که توزیع بار به صورت متعادل انجام شود و نیازی به دخالت دستی برای مدیریت شاردها نیست. این مزیت باعث می شود که توزیع بار در سراسر پایگاه داده کارآمدتر و عملکرد بهتر شود.

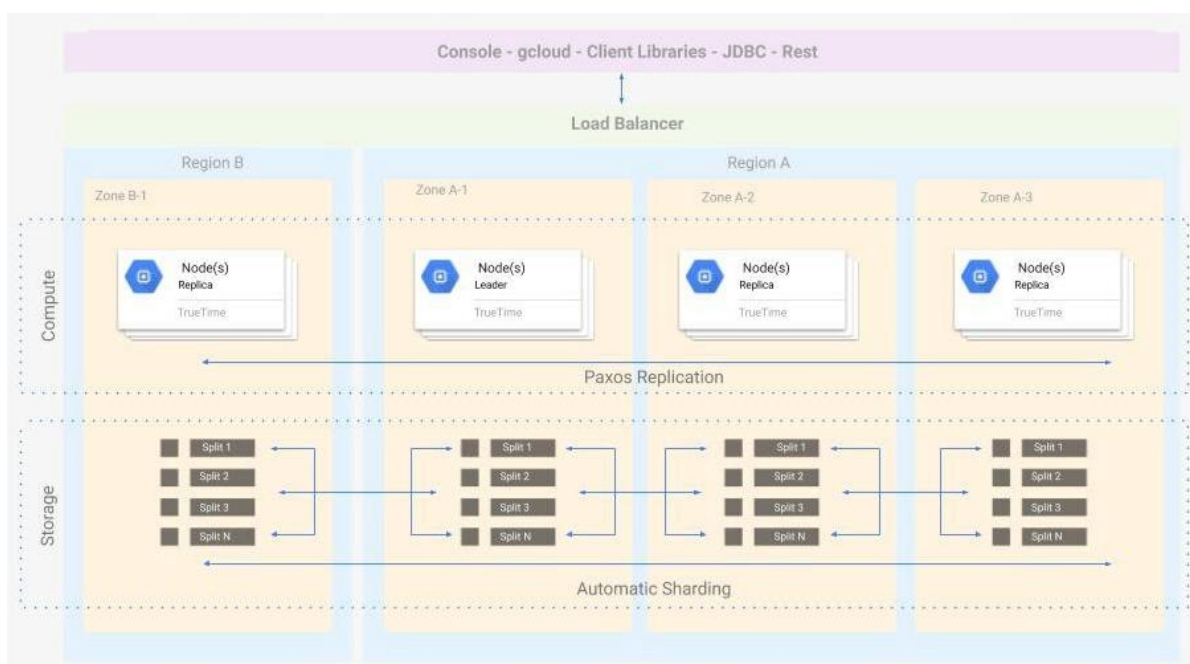
معماری Google Spanner از چندین جزء کلیدی تشکیل شده است:



۱. کلاینت‌ها (Clients): کلاینت‌ها درخواست‌های SQL را ارسال می‌کنند و از طریق API های Google Cloud Spanner متصل می‌شوند.

۲. سرورهای Frontend: این سرورها درخواست‌های کلاینت‌ها را مدیریت می‌کنند و آن‌ها را به گره‌های داده توزیع می‌کنند.

۳. True Time: سیستم ساعت‌های توزیع‌شده که برای هماهنگی دقیق زمانی استفاده می‌شود.



معماری پایگاه داده Google Spanner

پایگاه داده Cockroach DB

در مهندسی Cockroach DB یک پایگاه داده SQL توزیع شده است که برای فراهم کردن امکانات مقیاس پذیری، تحمل خطای بالا و ثبات طراحی شده است. این ابزار توسط Cockroach Labs توسعه داده شد و در سال ۲۰۱۵ به عنوان یک پروژه منبع باز منتشر شد. از آن زمان، به دلیل ویژگی‌های منحصر به فرد و سهولت در استفاده، بین توسعه دهندگان محبوبیت پیدا کرد. یکی از متمایزترین ویژگی‌های Cockroach DB توانایی آن در مقیاس‌پذیری افقی است. این بدان معنی است که با افزایش حجم کار در پایگاه داده، می‌توان گره‌های اضافی را به خوشه اضافه کرد تا بار افزایش یافته را مدیریت کند. Cockroach DB از یک معماری توزیع شده استفاده می‌کند تا تضمین می‌کند که داده‌ها همیشه در دسترس و منسجم هستند. معماری Cockroach DB مبتنی بر پایگاه داده Google Spanner است که برای ارائه ثبات و مقیاس پذیری برای برنامه‌های کاربردی توزیع شده در مقیاس بزرگ طراحی شده است. Cockroach DB علاوه بر امکانات مشابه Google Spanner، امکاناتی نظیر متعادل سازی خودکار داده‌ها، تحمل خطای

خودکار (automatic failover) و پشتیبانی از تراکنش‌های توزیع‌شده را به آن افزوده است. یکی از مزایای کلیدی Cockroach DB توانایی آن در ارائه سازگاری قوی در کل خوشه است. این بدان معناست که کاربران همیشه می‌توانند به داده‌هایی که از پایگاه داده بازیابی می‌کنند اطمینان کنند، حتی اگر به‌روزرسانی‌های همزمان در سیستم دریافت شود. سازگاری قوی برای بسیاری از برنامه‌ها ضروری است، به ویژه برنامه‌هایی که تراکنش‌های مالی یا داده‌های حساس را مدیریت می‌کنند.

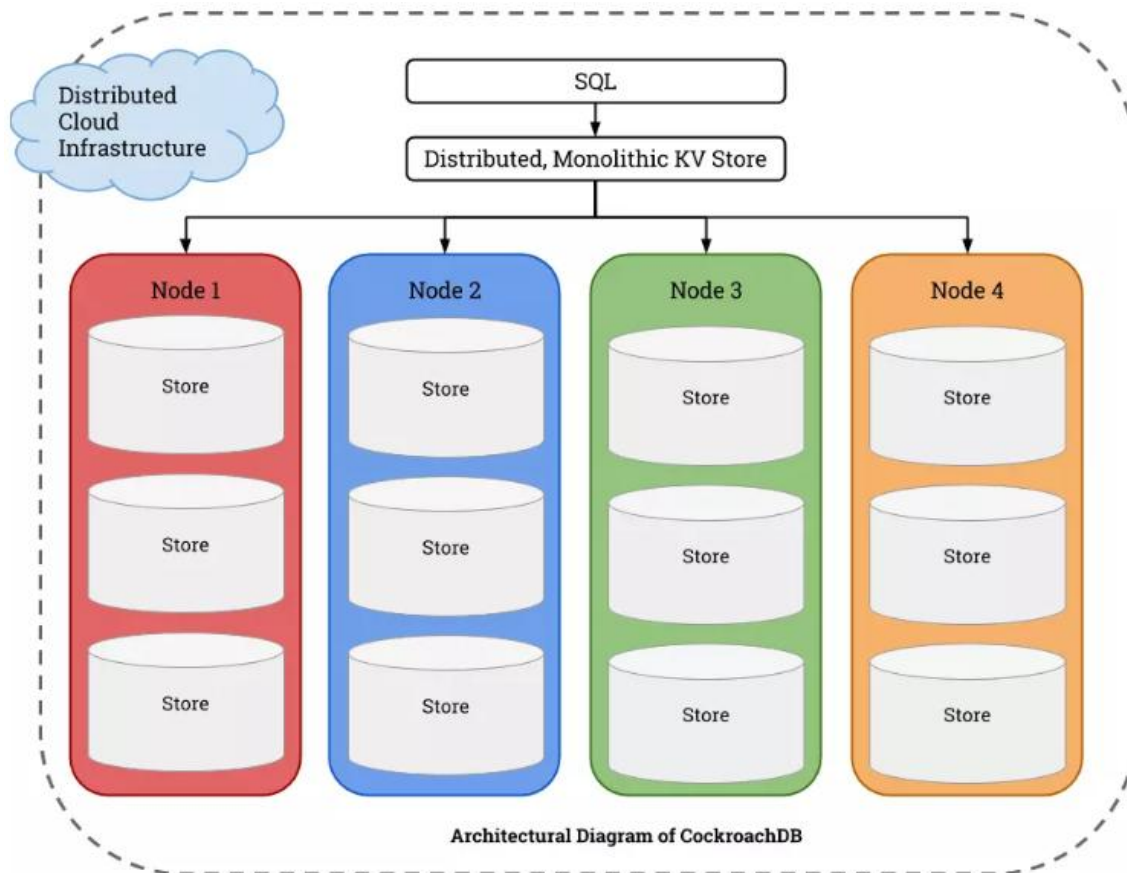
یکی دیگر از ویژگی‌های مهم Cockroach DB پشتیبانی آن از تراکنش‌های توزیع‌شده است. تراکنش‌ها راهی برای گروه‌بندی چندین عملیات مرتبط پایگاه داده به یک واحد منطقی است. در حین انجام تراکنش‌ها Cockroach DB اطمینان حاصل می‌کند که همه آنها با موفقیت انجام می‌شوند یا در صورت وجود خطایی در یکی از آن‌ها، همه به حالت قبل برگشت داده می‌شوند. تراکنش‌های توزیع‌شده به توسعه دهندگان این امکان را می‌دهد که برنامه‌های پیچیده‌ای بنویسند که چندین گره را در بر می‌گیرند و اطمینان حاصل می‌کنند که تمام عملیات‌ها با موفقیت انجام می‌شوند، حتی در صورت شکست.

Cockroach DB همچنین تعادل خودکار داده‌ها را در سراسر خوشه فراهم می‌کند و اطمینان می‌دهد که داده‌ها به طور مساوی توزیع شده‌اند و گره‌ها به طور کامل مورد استفاده قرار می‌گیرند. این ویژگی اضافه کردن گره‌های جدید را در صورت نیاز به خوشه آسان می‌کند، بدون اینکه نگران تعادل مجدد دستی یا انتقال داده باشید.

Cockroach DB علاوه بر ویژگی‌های اصلی خود، تعدادی ابزار و ویژگی‌های مفید دیگر را نیز در اختیار توسعه دهندگان قرار می‌دهد. به عنوان مثال، یک ویرایشگر داخلی SQL در آن تعبیه شده است که به توسعه دهندگان اجازه می‌دهد تا دستورات پایگاه داده را مستقیماً از خط فرمان اجرا کنند. همچنین از پشتیبان‌گیری توزیع‌شده پشتیبانی می‌کند.

Cockroach DB طوری طراحی شده است که استفاده از آن آسان باشد، حتی برای توسعه دهندگانی که با سیستم‌های توزیع‌شده یا مدیریت پایگاه داده آشنا نیستند. توسعه دهندگان می‌توانند بر روی ساخت برنامه‌های خود متمرکز شوند، بدون اینکه نگران زیرساخت‌ها و مدیریت پایگاه داده باشند. این ویژگی آن را به گزینه‌ای جذاب برای استارت‌آپ‌ها و کسب و کارهای کوچک تبدیل می‌کند که می‌خواهند بدون نیاز به هزینه‌های زیاد برای زیرساخت و طراحی معماری توزیع‌شده، برنامه‌های کاربردی مقیاس‌پذیر بسازند. با این حال، برخی محدودیت‌ها برای Cockroach DB وجود دارد. برای مثال، ممکن است برای برنامه‌هایی که به تأخیر بسیار کم نیاز دارند، مناسب نباشد، زیرا در سیستم‌های توزیع‌شده، همیشه مقداری تأخیر در اثر ارتباطات بوجود می‌آید.

Cockroach DB به علت سهولت در استفاده و امکانات مفیدی که ارائه می‌کند، روزبه‌روز در حال فراگیرتر شدن است. همچنین با توسعه و گسترش مداوم خود احتمالاً در سال‌های آینده نقش مهمی را در دنیای پایگاه‌های داده توزیع‌شده ایفا خواهد کرد.



معماری پایگاه داده Cockroach DB

پایگاه داده Amazon Aurora

Amazon Aurora یک سرویس پایگاه داده رابطه‌ای (RDBMS) کاملاً مدیریت شده است که توسط Amazon Web Services (AWS) ارائه می‌شود. این پایگاه داده به گونه‌ای طراحی شده که سرعت و قابلیت اطمینان پایگاه‌های تجاری پیشرفته مانند Oracle و SQL Server را با سادگی و هزینه پایین MySQL و PostgreSQL ترکیب کند.

Amazon Aurora برای کاربردهای حساس به تأخیر، برنامه‌های مقیاس‌پذیر و برنامه‌های کاربردی آنلاین ایده‌آل است. برخلاف پایگاه‌های داده سنتی، Aurora به طور کامل با AWS Cloud یکپارچه شده و از امکانات مقیاس‌پذیری خودکار، بازیابی سریع، در دسترس بودن بالا (High Availability) و امنیت قوی برخوردار است. از جمله ویژگی‌های آن می‌توان به موارد زیر اشاره کرد.

- سازگاری با MySQL و PostgreSQL: Amazon Aurora کاملاً با MySQL و PostgreSQL سازگار است، به این معنی که برنامه‌های موجود با کمترین تغییرات می‌توانند به Aurora منتقل شوند. کاربران می‌توانند از همان ابزارهای SQL، ORM ها) مانند Hibernate) و کتابخانه‌های توسعه نرم‌افزار که برای MySQL و PostgreSQL استفاده می‌کنند، برای Aurora نیز استفاده کنند.

- معماری ذخیره سازی توزیع شده : برخلاف بسیاری از پایگاه های داده سنتی که از ذخیره سازی محلی استفاده می کنند، Aurora از ذخیره سازی توزیع شده استفاده می کند. داده ها به صورت خودکار به ۶ نسخه (کپی) در سه منطقه در دسترس (Availability Zones) تکرار می شوند. در صورت بروز خطا در هر کپی از داده ها، به طور خودکار بازسازی داده ها انجام می شود، که باعث افزایش در دسترس بودن (High Availability) می شود.

- سرعت و عملکرد بالا : عملکرد ۵ Aurora برابر سریع تر از MySQL و ۳ برابر سریع تر از PostgreSQL است. این پایگاه داده از خواندن موازی (Parallel Read) استفاده می کند، به این معنی که می تواند درخواست های خواندن را از چندین کپی داده به طور همزمان اجرا کند. Aurora از مکانیزم Write-Ahead Logging (WAL) استفاده می کند که نوشتن داده ها را کارآمدتر می سازد.

- پشتیبانی از مقیاس پذیری خودکار (Auto-scaling) : Aurora به طور خودکار مقیاس پایگاه داده را بر اساس نیاز کاری (Workload) افزایش یا کاهش می دهد. این پایگاه داده می تواند از حجم های کوچک (۱۰ گیگابایت) تا ۱۲۸ ترابایت گسترش یابد. در صورت افزایش بار کاری، گره های خواندن و نوشتن بیشتری اضافه می شوند تا درخواست های خواندن/نوشتن به طور همزمان انجام شوند.

- در دسترس بودن بالا (High Availability) و تحمل خطا : معماری چند منطقه ای Aurora به این معنی است که داده ها در سه منطقه در دسترس (Availability Zones) در AWS توزیع شده اند. در صورت بروز خطا در یک منطقه، پایگاه داده به طور خودکار به نسخه های دیگر منتقل می شود. Aurora همچنین از Failover خودکار پشتیبانی می کند؛ به این معنی که اگر سرور اصلی خراب شود، سرور پشتیبان به طور خودکار کنترل را در دست می گیرد.

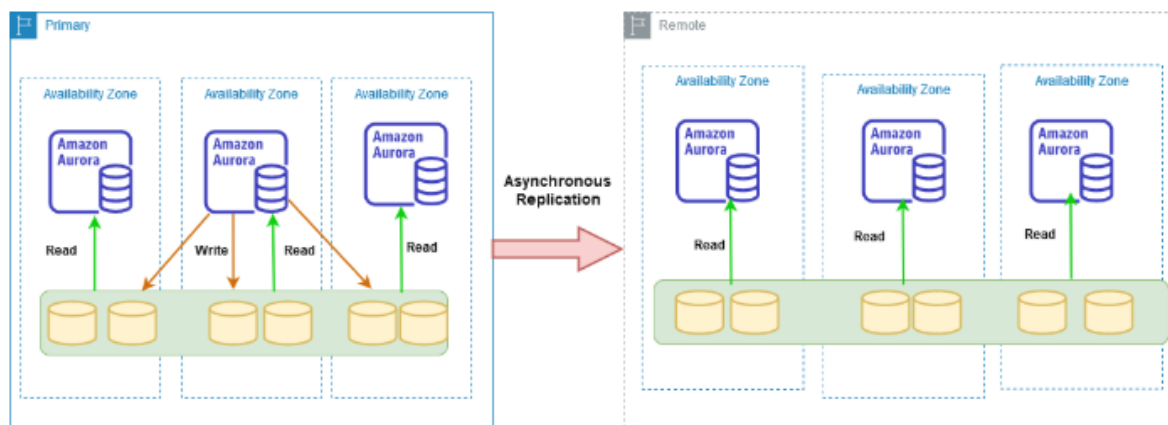
- پشتیبانی از نقاط بازیابی (Point-in-time Recovery) : Aurora امکان بازیابی داده ها به نقطه ای در زمان خاص را فراهم می کند. شما می توانید هرگونه تغییر ناخواسته در داده ها (مانند حذف اشتباهی) را به سرعت بازیابی کنید.

- رمزنگاری و امنیت داده ها : Aurora از رمزنگاری داده ها در حالت سکون (At-rest) و در حالت انتقال (In-transit) پشتیبانی می کند. رمزنگاری داده ها از طریق AWS Key Management Service (KMS) انجام می شود. کنترل های امنیتی مبتنی بر نقش، احراز هویت چندعاملی (MFA) و اتصال امن به پایگاه داده با استفاده از SSL/TLS نیز ارائه می شوند.

معماری Aurora از دو بخش اصلی تشکیل شده است:

- لایه محاسباتی (Compute Layer) : این بخش درخواست های خواندن و نوشتن SQL را پردازش می کند. برای خواندن، داده ها از نسخه های کپی شده در منطقه های مختلف بازیابی می شوند. برای نوشتن، اطلاعات به ذخیره سازی توزیع شده ارسال می شوند و در ۶ نسخه در سه منطقه تکرار می شوند.

- لایه ذخیره سازی (Storage Layer) : این لایه مسئول ذخیره سازی رکوردهای توزیع شده و شارد شده است. داده ها در سه منطقه AWS توزیع شده اند و هر داده دارای ۶ کپی (Replica) است. اگر یک نسخه از داده آسیب ببیند، به طور خودکار یک کپی دیگر جایگزین آن می شود.



معماری پایگاه داده Aurora

مقایسه سه پایگاه داده

برای مقایسه Google Spanner، Cockroach DB و Amazon Aurora، معیارهای زیر در نظر گرفته شده است: سازگاری ACID، مقیاس پذیری، در دسترس بودن، تاخیر و موارد استفاده.

Amazon Aurora	Google Spanner	Cockroach DB	پایگاه داده / معیار
نسبتا	بله	بله	سازگاری ACID
عمودی	افقی	افقی	مقیاس پذیری
پایین	پایین (با هماهنگی زمانی)	پایین	تاخیر
بسیار بالا	بسیار بالا	بسیار بالا	در دسترس بودن
برنامه های تجاری و آنلاین	سیستم های مالی و بانکی	سیستم های توزیع شده	موارد استفاده



هر یک از پایگاه‌های داده توزیع‌شده مورد بررسی در این مقاله، مزایا و معایب خود را دارند. Cockroach DB برای شرکت‌هایی که به شفافیت و قابلیت‌های مقیاس‌پذیری افقی نیاز دارند، مناسب است. Google Spanner به دلیل توانایی مدیریت هماهنگی جهانی، بهترین گزینه برای بانک‌ها و مؤسسات مالی است. Amazon Aurora برای برنامه‌های کاربردی آنلاین که به پایگاه داده‌های سازگار با MySQL یا PostgreSQL نیاز دارند، انتخاب خوبی محسوب می‌شود.

نتیجه‌گیری

این مقاله به بررسی و مقایسه سه پایگاه داده توزیع‌شده Cockroach DB، Google Spanner و Amazon Aurora پرداخت. هر یک از این سیستم‌ها برای کاربردهای خاصی بهینه شده‌اند. Cockroach DB برای شرکت‌هایی که به شفافیت و مقیاس‌پذیری افقی نیاز دارند، مناسب است. Google Spanner برای هماهنگی جهانی و مدیریت تراکنش‌های مالی انتخاب خوبی است و Amazon Aurora گزینه‌ای مناسب برای برنامه‌های آنلاین و تجاری است.

منابع

۱. IEEE Xplore. A Systematic Literature Review on Cloud Computing Security.
۲. Springer. Distributed Databases in Modern Applications.
۳. ACM Digital Library. Cockroach DB: Scaling SQL Databases Globally.
۴. Google Cloud Documentation: Spanner Overview.
۵. Amazon Web Services (AWS) Whitepaper: Amazon Aurora Overview.
۶. Database Trends and Applications: Distributed Databases in ۲۰۲۴.
۷. ResearchGate. Advanced Concepts in Distributed Databases.
۸. IEEE Xplore. New Trends in Distributed Database Systems.
۹. Springer. Cloud-Native Databases and Their Implications.
۱۰. ACM Transactions on Database Systems. Modern Techniques in Database Sharding.



Comparison of distributed databases: CockroachDB, Google Spanner and Amazon Aurora

Mohammadjavad Hoseinpour

Member of the Faculty of Computer Engineering, Islamic Azad University, Estehban Branch,
Estehban, Iran

Fatemeh Zare

Master's student, Department of Computer Engineering, Islamic Azad University, Estehban Branch,
Estehban, Iran

Abstract

In recent years, the need for distributed databases to support high scalability and continuous availability has increased. CockroachDB, Google Spanner and Amazon Aurora are three leading systems in this field. This paper compares these three systems in terms of architecture, ACID compatibility, scalability, latency and performance. The results show that each system is suitable for specific use cases. For example, Spanner is better suited for banking systems due to its strong ACID compatibility, while CockroachDB works better for horizontal scalability.

Keywords: Database, Distributed Database, Google Spanner, CockroachDB, Amazon Aurora