عنوان: **NOSQL قسمت دوم**
نویسنده: حمید سامانی
تاریخ: ۱۳۹۱/۱۱/۲۶ ۹:۲۵
آدرس: www.dotnettips.info
گروه‌ها: NoSQL, Database, پایگاه‌داده, نوسی‌کوال, نواس‌کیوال, key-value, کلید-مقدار

در مطلب قبلی با تعاریف سیستم‌های NoSQL آشنا شدیم و به طور کلی ویژگی‌های یک سیستم NoSQL را بررسی کردیم.

در این مطلب دسته‌بندی کلی و نوع ساختار داده‌ای این سیستم‌ها و بررسی ساده‌ترین آنها را مرور می‌کنیم.

در حالت کلی پایگاهای داده NoSQL به ۴ دسته تقسیم می‌شوند که به ترتیب پیچیدگی ذخیره‌سازی داده‌ها عبارتند از:
Key/Value Store Databases

Document Databases
Graph Databases
Column Family Databases
در حالت کلی در پایگاه‌های‌داده NoSQL داده‌ها در قالب KEY/VALUE (کلید/مقدار) نگه‌داری می‌شوند ، به این صورت که مقادیر توسط کلید یکتایی نگاشت شده و ذخیره می‌شوند، هر مقدار صرفا توسط همان کلید نگاشت شده قابل بازگردانی می‌باشد و راهی جهت دریافت مقدار بدون دانستن کلید وجود ندارد . در این ساختارداده منظور از مقادیر، داده‌های اصلی برنامه هستند که نیاز به نگه‌داری دارند و کلیدها نیز رشته‌هایی هستند که توسط برنامه‌نویس ایجاد می‌شوند.
به دلیل موجود بودن این نوع ساختار داده‌ای در اکثر کتابخانه‌های زبان‌های برنامه‌نویسی (به عنوان مثال پیاده‌سازی‌های مختلف اینترفیس Map شامل HashMap ، HashTable و موارد دیگر در کتابخانه‌های JDK) این نوع ساختار برای اکثر برنامه‌نویسان آشنا بوده و فراگیری آن نیز ساده می‌باشد.

بدیهی است که اعمال فرهنگ داده‌ای (درج ، حذف ، جستجو) در این سیستم به دلیل اینکه داده‌ها به صورت کلید/مقدار ذخیره می‌شوند دارای پیچیدگی زمانی O(1) می‌باشد که بهینه‌ترین حالت ممکن به لحاظ طراحی می‌باشد. همان‌گونه که مستحضرید در الگوریتم‌هایی که دارای پیچیدگی زمانی با مقدار ثابت دارند کم یا زیاد بودن داده‌ها تاثیری در کارایی الگوریتم نداشته و همواره با هر حجم داده‌ای زمان ثابتی جهت پردازش نیاز می‌باشد.

:Key/Value Store Databases

این سیستم ساده‌ترین حالت از دسته‌بندی‌های NoSQL می‌باشد ، به طور کلی جهت استفاده در سیستم‌هایی است که داده‌ها متمایز از یکدیگر هستند و اصولا Availability و یا در دسترس بودن داده‌ها نسبت به سایر موارد نظیر پایائی اهمیت بالاتری دارد.

از موارد استفاده این گونه سیستم‌ها به موارد زیر می‌توان اشاره کرد:
در پلتفرم‌های اشتراک گذاری داده‌ها . هدف کلی صرفا هندل کردن آپلود محتوی (باینری) و به صورت همزمان بروز کردن در سمت دیگر می‌باشد.(اپلیکیشنی مانند اینستاگرام را تصور کنید) در اینگونه نرم‌افزارها با تعداد بسیار زیاد کاربر و تقاضا، استفاده از این نوع پایگاه داده به مراتب کارایی و سرعت را بالاتر می‌برد. و با توجه به عدم پیش‌بینی حجم داده‌ها یکی از ویژگی‌های این نوع پایگاه داده تحت عنوان Horizontal Scaling مطرح می‌شود که در صورت Overflow شدن سرور، داده‌ها را به سمت سرور دیگری می‌توان هدایت کرد وبدون مشکل پردازش را ادامه داد ، این ویژگی یک وجه تمایز کارایی این سیستم با سیستم‌های RDBMS می‌باشد که جهت مقابله با چنین وضعیتی تنها راه پیش‌رو بالا بردن امکانات سرور می‌باشد و به طور کلی داده‌ها را در یک سرور می‌توان نگه‌داری کرد (البته راه‌حل‌هایی نظیر پارتیشن کردن و غیره وجود دارد که به مراتب پیچیدگی و کارایی کمتری نسبت به Horizontal Scaling در پایگاه‌های داده NoSQL دارد.)

برای Cache کردن صفحات بسیار کارا می‌باشد ، به عنوان مثال می‌توان آدرس درخواست را به عنوان Key در نظر گرفت و مقدار آن را نیز معادل JSON نتیجه که توسط کلاینت پردازش خواهد شد قرار داد.

یک نسخه کپی شده از توئیتر که کاملا توسط این نوع پایگاه داده پیاده شده است نیز از این آدرس قابل مشاهده است. این برنامه به زبان‌های ruby , php و java نوشته شده است و سورس نیز در مخارن github می‌جود می‌باشد. (یک نمونه پیاده سازی ایده‌آل جهت آشنایی با نحوه‌ی مدیریت داده‌ها در این نوع پایگاه داده)

از پیاده‌سازی‌های این نوع پایگاه داده به موارد زیر می‌توان اشاره کرد:

Amazon SimpleDB

Memcached

Oracle Key/value Pair

Redis

هر یک از پیاده‌سازی‌ها دارای ویژگی‌های مربوط به خود هستند به عنوان مثال Memcached داده‌ها را صرفا در DRAM ذخیره می‌کند که نتیجه‌ی آن Volatile بودن داده‌ها می‌باشد و به هیچ وجه از این سیستم جهت نگهداری دائمی داده‌ها نباید استفاده شود. از طرف دیگر Redis داده‌ها را علاوه بر حافظه اصلی در حافظه جانبی نیز ذخیره می‌کند که نتیجه‌ی آن سرعت بالا در کنار پایائی می‌باشد.

همان‌گونه که در تعریف کلی عنوان شد یکی از ویژگی‌های این سیستم‌ها متن‌باز بودن انها می‌باشد که نتیجه‌ی آن وجود پیاده‌سازی‌های متنوع از هر کدام می‌باشد ، لازم است قبل از انتخاب هر سیستم به خوبی با ویژگی‌های اکثر سیستم‌های محبوب و پراستفاده آشنا شویم و با توجه به نیاز سیستم را انتخاب کنیم.

در مطلب بعدی با نوع دوم یعنی Document Databases آشنا خواهیم شد.

نظرات خوانندگان

نویسنده: مجید هزاری
تاریخ: ۱۳۹۱/۱۱/۲۸ ۱۵:۴۷

عالی است.
متشکرم.

نویسنده: احمد ولی پور
تاریخ: ۱۳۹۱/۱۱/۲۸ ۱۷:۵۵

یه سوال برام پیش اومده:
با رایج شدن nosql پایگاه داده هایی مثل Oracle یا Sql Server چی میشن؟

نویسنده: مجید هزاری
تاریخ: ۱۳۹۱/۱۱/۲۸ ۱۹:۵۰

اینها تداخلی با یکدیگر ندارند.
NOSQL تنها برای رفع نیاز هایی ظهور کرده است که RelDB در آنها ضعیف بوده. همانطور که NoSQL در زمینه هایی که RelDB قوی است ضعیف عمل خواهد کرد.
(البته من کاملا مختصر گفتم)

نویسنده: حمید سامانی
تاریخ: ۱۳۹۱/۱۱/۲۸ ۲۰:۱۷

در حالت کلی هرکدام از پایگاه‌داده‌ها بسته به نیاز استفاده می‌شن ، توی برنامه‌های اینترپرایز وبی مفهوم Polyglot Persistence مطرحه (که می‌شه اونو نگه‌داری یا ذخیره سازی چند زبانی ترجمه کرد) که می‌گه توی یک سیستم از چندین نوع پایگاه داده ~~می‌شه~~ (باید) استفاده کرد. به عنوان مثال برای نگه‌داری داده‌هایی جهت گزارش گیری و یا ایجاد Transactionها بهترین گزینه همان سیستم‌های RDBMS هستند ، در مطالب آتی به این موضوع اشاره بیشتری خواهم کرد ، مارتین فویلر در این مطلب مفهوم Polyglot Persistence را به خوبی توضیح داده‌اند.

نویسنده: saremi
تاریخ: ۱۳۹۲/۱۱/۲۹ ۱۷:۲۲

سلام
می خواستم بپرسم bucket در key value store دقیقا چیه؟
بعد خیلی از جاها راجع به hash table و hash code هم مطالبی گفته اند. آیا منظور فقط hash کردن کلید است یا فرآیند پیچیده‌تر از این حرفاست؟