1. **GIỚI THIỆU**
2. **Tổng quan về cơ sở dữ liệu NoSQL**
3. **Tổng quan kỹ năng**

## **Mô hình dữ liệu kiến trúc**

### **Tùy chọn mô hình hóa dữ liệu**

### **Mô hình Sharding**

### **Tính nhất quán và mô hình độ bền**

**Consistency** đảm bảo rằng một giao dịch không bao giờ được thông qua cơ sở dữ liệu của bạn trong tình trạng dở dang. Tính chất này, hoặc là tạo ra toàn bộ trạng thái mới hoặc rollback tất cả các xử lý để về trạng thái ban đầu, nhưng không bao giờ thông qua cơ sở dữ liệu trong trạng thái dở dang.

**Durability** đảm bảo rằng cơ sở dữ liệu sẽ theo dõi các thay đổi cấp phát trong một cách mà các máy chủ có thể phục hồi từ một sự kết thúc bất thường. Tính chất này đảm bảo rằng trong trường hợp thất bại hay dịch vụ khởi động lại các dữ liệu có sẵn trong  trước khi gặp lỗi.

### **Mô hình lập trình**

### **Mô hình bảng**

### **Truy cập dữ liệu bảng qua SQL**

### **Làm việc với tài liệu JSON**

**Truy cập dữ liệu bảng qua SQL**

Đối với các truy vấn không phù hợp với kiểu truy cập từ khóa đơn giản/giá trị đơn giản hoặc quét phạm vi đơn giản bằng khóa phụ, Cơ sở dữ liệu Oracle NoSQL cung cấp quyền truy cập SQL gốc vào dữ liệu. Mặc dù Cơ sở dữ liệu Oracle NoSQL cung cấp một phương ngữ SQL khá phong phú, SQL này có thể được coi là một tập hợp con đơn giản hơn của SQL mạnh hơn và toàn diện hơn do Oracle RDBMS cung cấp. Đối với những ứng dụng yêu cầu mức truy cập SQL này, Cơ sở dữ liệu Oracle NoSQL được tích hợp với Oracle RDBMS, cho phép RDBMS truy vấn dữ liệu trực tiếp từ các bảng Cơ sở dữ liệu Oracle NoSQL.

Oracle NoSQL của SQL là khái niệm về khả năng truy vấn SQL đối với dữ liệu phi quan hệ như mảng và bản đồ. Phương ngữ Một lĩnh vực khác biệt đáng chú ý giữa các phương ngữ Cơ sở dữ liệu Oracle RDBMS và SQL trong Cơ sở dữ liệu Oracle NoSQL cung cấp các cấu trúc mạnh mẽ để cắt qua các mảng và bản đồ cũng như cung cấp các biểu thức lọc trên các cấu trúc này. Các ví dụ dưới đây minh họa một tập hợp con nhỏ của khả năng này.

1. Trả về họ và tên của tất cả người dùng ở thành phố New York: chọn FirstName, lastName từ người dùng có địa chỉ.phones[$element.city = “New York”]

2. Trả về họ và tên của tất cả người dùng có địa chỉ nhà riêng trong mã zip 94107 và sắp xếp kết quả theo thuộc tính lastName: chọn FirstName, lastName từ người dùng trong đó places.phones[$element.zip= 94107] và address.phones [$element.attrType = “home”] sắp xếp by lastName

3. Hiển thị trang đầu tiên (25 kết quả mỗi trang) của tất cả người dùng ở New Haven, Newark và Thành phố New York: chọn \* từuser where address.phones[$element.city = “New York”] hoặc Cơ sở dữ liệu Oracle NoSQL Tổng quan 8 địa chỉ.phones[$element.city = “New Haven”] hoặc address.phones[$element.city = “Newark”] đặt hàng theo LastName giới hạn 25 offset 1

**Làm việc với Tài liệu JSON**

JSON đã phát triển đáng kể trong vài năm qua dưới dạng định dạng dữ liệu thuận tiện cho các ứng dụng web, trả về kết quả từ các cuộc gọi API dịch vụ và trao đổi dữ liệu giữa các ứng dụng.Là một kiểu dữ liệu tự mô tả, dữ liệu được định dạng JSON cung cấp tính linh hoạt cao nhất cho những ứng dụng muốn lưu trữ nội dung đặc biệt mà không cần phải chỉ định trước lược đồ.

Các lược đồ cố định được lưu trữ một lần, trong khi mọi tài liệu JSON chứa một bản sao của lược đồ.

Cơ sở dữ liệu Oracle NoSQL cung cấp một sự kết hợp độc đáo của các bảng lược đồ cố định cũng như hỗ trợ không cần lược đồ đặc biệt cho các tài liệu JSON. Các nhà phát triển có thể chọn phần nào của ứng dụng sẽ tận dụng tính linh hoạt của JSON không cần lược đồ và những phần nào của ứng dụng nên tối ưu hóa dung lượng lưu trữ và tính toán để đổi lấy lược đồ cố định.

Phiên bản lược đồ cố định từ ví dụ trước của chúng ta có thể được biểu diễn dưới dạng JSON không có lược đồ bên dưới:

Statement result res = KVStore.executeSync(“CREATE TABLE user(id INTEGER, userData JSON primary key(id)))”);

và dữ liệu JSON có thể được chèn bằng cách sử dụng:

Row r = Table.createRow().put(“id”, 123456).put("userData", FieldValueFactory.createValueFromJson(

“{

“firstName” : “John”,

“lastName” : “Doe”

"addresses": [ {

“street” : “127 Spring St”, “city” : “New York”, “state” : “NY”,

“zip” : 10012, ”addrType” : “work”},

{“street” : “625 Ridgewood Rd”, “city” : “Paramus”, “state” : “NJ”,

“zip” : 07675, ”addrType” : “home”}

] }”);

));

Dữ liệu JSON cũng có thể được truy vấn bằng cách sử dụng cùng một ngôn ngữ SQL hoạt động trên Bảng. Ngoài ra, đối với các tài liệu JSON đặc biệt , một số toán tử mới và biểu thức "trường hợp" được giới thiệu để hỗ trợ xử lý các tài liệu có thể có định dạng rời rạc hoặc không xác định.

1. Tìm tất cả người dùng ở thành phố New York. Trong trường hợp này, nếu tài liệu có một mảng địa chỉ thì hãy áp dụng vị ngữ cho mảng, nếu không, hãy áp dụng vị ngữ trực tiếp cho một thuộc tính được gọi là "thành phố" trong chính tài liệu.

select

firstName,

lastName

from

User u

where

(case when u.addresses instanceof ARRAY then

u.addresses[$element.city = “New York”] else

u.city = “New York” end)

1. Tìm tất cả người dùng ở Thành phố New York khi tài liệu JSON chứa thuộc tính thành phố, trong tài liệu dành cho vị ngữ truy vấn

select \*

from User u

where

(case when exists(u.address.city) then u.address.city = “New York”

Việc kết hợp Cơ sở dữ liệu Oracle NoSQL vào các ứng dụng rất đơn giản. Các API cho các thao tác Tạo, Đọc, Cập nhật và Xóa (CRUD) cơ bản, cũng như truy cập SQL, được đóng gói trong một tệp jar duy nhất. Các ứng dụng có thể sử dụng các API hoặc truy vấn SQL từ một hoặc nhiều quy trình máy khách truy cập vào quy trình máy chủ cơ sở dữ liệu Oracle NoSQL Database độc lập, giảm bớt nhu cầu thiết lập cấu hình đa hệ thống để phát triển và thử nghiệm ban đầu.

**Khả năng phục hồi thất bại**

Cơ sở dữ liệu Oracle NoSQL đưa ra khái niệm về các vùng khả dụng dưới dạng các thùng chứa cách ly lỗi. Vùng khả dụng có thể ở dạng giá đỡ máy chủ, công tắc trên cùng của giá đỡ, sàn của tòa nhà hoặc toàn bộ trung tâm dữ liệu. Các khu vực cũng được mô tả là chính hoặc phụ, có các khu vực chính dành cho bầu cử tổng thể cũng như xác nhận số đại biểu cần thiết.Cơ sở dữ liệu Oracle NoSQL tiếp tục cho phép cấu hình bất kỳ số vùng chính và phụ nào. Sau khi cấu hình, các bản sao cho từng phân đoạn trong kho lưu trữ dữ liệu được bố trí trên các vùng này sao cho lỗi của bất kỳ vùng nào sẽ hạn chế (hoặc chứa) lỗi này để mỗi phân đoạn trong kho dữ liệu luôn duy trì đủ số đại biểu cần thiết.

Cơ sở dữ liệu Oracle NoSQL triển khai một Dịch vụ quản trị riêng có tính khả dụng cao. Nhất quán với triết lý “Không có điểm lỗi duy nhất” của Cơ sở dữ liệu Oracle NoSQL, hoạt động liên tục của một cài đặt không phụ thuộc vào tính khả dụng của Dịch vụ quản trị. Do đó, cả cơ sở dữ liệu và Dịch vụ quản trị vẫn khả dụng trong quá trình thay đổi cấu hình.

**Ngành kiến trúc**

Chúng tôi trình bày kiến trúc Cơ sở dữ liệu Oracle NoSQL bằng cách theo dõi quá trình thực hiện thao tác ghi thông qua các thành phần logic của hệ thống và sau đó thảo luận về cách các thành phần đó ánh xạ tới hoạt động của phần cứng và phần mềm thực tế. Chúng tôi sẽ tạo một bản ghi với khóa “Katana” và một đối tượng JSON là giá trị “{“a”: “foo”}”. Hình 3 mô tả lời gọi phương thức putIfAbsent(“Katana”, “{“a”: “foo”}

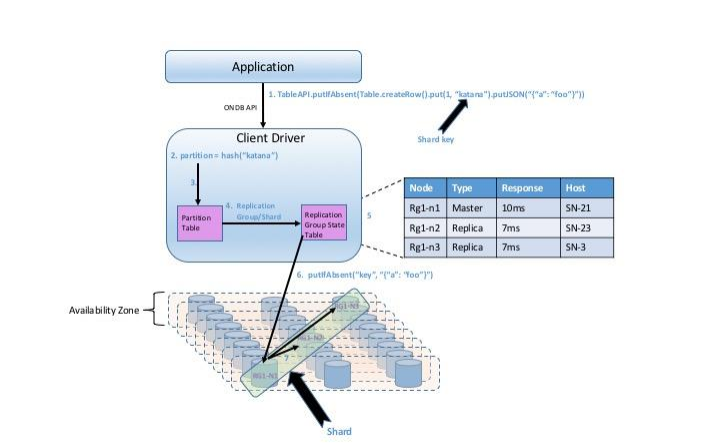


Figure 3 – Architecture - Oracle NoSQL Database – Write Request Flow

Ứng dụng đưa ra phương pháp putIfAbsent trình điều khiển khách hàng (bước 1). Trình điều khiển máy khách băm khóa “Katana” để chọn một trong số các phân vùng (bước 2). Mỗi phân vùng được gán cho một nhóm sao chép cụ thể (phân đoạn). Trình điều khiển tham khảo bảng phân vùng (bước 3) để ánh xạ số phân vùng vào một phân đoạn.

Một phân đoạn bao gồm một số (có thể định cấu hình) số lượng nút sao chép, và mỗi nút sao chép nằm trong một vùng sẵn sàng riêng biệt (để ngăn chặn lỗi). Số lượng nút sao chép trong một phân đoạn quyết định số lượng lỗi mà từ đó hệ thống có khả năng phục hồi; một hệ thống có ba nút trên mỗi phân đoạn có thể chịu được hai lỗi trong khi tiếp tục phục vụ các yêu cầu đọc dịch vụ. Khả năng chịu đựng các lỗi ghi của nó dựa trên chính sách độ bền được yêu cầu trong lệnh gọi API ghi. Nếu ứng dụng không yêu cầu số người tham gia xác nhận một lần ghi, thì hệ thống cũng chỉ có thể chịu được tối đa hai lỗi ghi. Một nhóm năm nút có thể chịu được tối đa đến bốn lỗi đối với mỗi lần đọc và tối đa hai lần lỗi đối với hai lần ghi, ngay cả khi ứng dụng yêu cầu chính sách độ bền và yêu cầu phần lớn các trang web xác nhận thao tác ghi.

Được cung cấp một phân đoạn, Trình điều khiển máy khách tiếp theo sẽ tham khảo Bảng trạng thái nhóm sao chép (RGST) (bước 4). Đối với mỗi phân đoạn, RGST chứa thông tin về từng nút sao chép bao gồm nhóm các nút sao chép trong phân đoạn (bước 5). Dựa trên thông tin trong RGST, chẳng hạn như danh tính của máy chủ và tải trên các nút khác nhau trong nhóm sao chép, Trình điều khiển Máy khách chọn nút để gửi yêu cầu và chuyển tiếp yêu cầu đến nút thích hợp (bước 6 ). Trong trường hợp này, vì chúng tôi đang thực hiện thao tác ghi, nên yêu cầu phải đi đến nút chính.

Sau đó, nút sao chép sẽ áp dụng thao tác. Trong trường hợp putIfAbsent, nếu khóa tồn tại, thao tác không có có hiệu lực và trả về lỗi, cho biết rằng mục nhập được chỉ định đã có trong cửa hàng. Nếu chìa khóa không tồn tại, nút sao chép sẽ thêm cặp khóa/giá trị vào cửa hàng và sau đó truyền cặp khóa/giá trị mới tới các nút khác trong nhóm sao chép (bước 7).

**Cài đặt**

Một bản cài đặt Cơ sở dữ liệu Oracle NoSQL bao gồm hai phần chính: Trình điều khiển máy khách và bộ sưu tập các Nút lưu trữ. Như được hiển thị trong Hình 3, trình điều khiển máy khách triển khai bản đồ phân vùng và RGST, trong khi các nút lưu trữ triển khai các nút sao chép bao gồm các phân đoạn. Trong phần này, chúng ta sẽ xem xét kỹ hơn từng thành phần này

**Nút lưu trữ**

Nút lưu trữ (SN) thường là một máy vật lý có bộ lưu trữ liên tục cục bộ, dạng đĩa hoặc trạng thái rắn, CPU có một hoặc nhiều lõi, bộ nhớ và địa chỉ IP. Hệ thống có nhiều nút lưu trữ hơn sẽ cung cấp thông lượng tổng hợp hoặc dung lượng lưu trữ lớn hơn so với hệ thống có ít nút hơn và các hệ thống có mức độ sao chép lớn hơn trong các nhóm sao chép có thể giảm độ trễ yêu cầu đối với các cài đặt có mức độ sao chép nhỏ hơn. Ngoài ra, nhiều SN hơn dẫn đến tính khả dụng cao hơn trên toàn bộ hệ thống.