**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP HỒ CHÍ MINH**

**KHOA : HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**



**ĐỒ ÁN**

**HỌC PHẦN : HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI: NOSQL-DATABASE-498041**

Nhóm thực hiện **: Nhóm 7**

Lớp **: 09\_ĐHCNTT\_03**

Giảng viên hướng dẫn **:** **Ths. Phạm Trọng Huynh**

**Tp HCM, tháng 12 năm 2022**

**O R A C L E W H I T P A P E R | J U N E , 2 0 1 8**

**CƠ SỞ DỮ LIỆU ORACLE NOQUERY**

**NHANH CHÓNG-ĐÁNG TIN CẬY-DỰ ĐOÁN ĐƯỢC**

**NỘI DUNG**

Được thúc đẩy bởi sáng kiến WeLoveStartups của Oracle Pháp, chúng tôi đã tăng cường đáng kể khả năng Big Data của mình bằng cách sử dụng Dịch vụ Dữ liệu Oracle REST để tận dụng Cơ sở dữ liệu Oracle NoSQL, Phiên bản Doanh nghiệp. Chỉ có Oracle mới có thể giúp chúng tôi đạt được cơ sở hạ tầng Big Data đầy đủ chức năng chỉ trong vài giờ.

0WASSEL GUERBAA, Giám đốc điều hành

CEO BUSIT SAS

MỤC LỤC

[Giới thiệu 4](#_Toc121053989)

[Tổng quan về Oracle NoSQL Database 4](#_Toc121053990)

[TỔNG QUAN VỀ KỸ THUẬT: 8](#_Toc121053991)

[Architectural Data Model (Mô hình dữ liệu Mô hình): 8](#_Toc121053992)

[Data Modeling Options (Tùy chọn mô hình hóa dữ liệu): 8](#_Toc121053993)

[Sharding Model (MÔ HÌNH CHIA NHỎ) 10](#_Toc121053994)

[Consistency and Durability Model (Tính nhất quán và độ bền) : 10](#_Toc121053995)

[Programming Models (Mô hình lập trình): 11](#_Toc121053996)

[The Table Model (Mô hình bảng) : 12](#_Toc121053997)

[Accessing Table Data via SQL (Truy cập bảng dữ liệu qua SQL) : 13](#_Toc121053998)

[Working with JSON Documents ( LÀM VIỆC VỚI TÀI LIỆU JSON) 14](#_Toc121053999)

[Failure Resiliency (KHẢ NĂNG CHỐNG LỖI) 17](#_Toc121054000)

[Architecture (Mô hình) 17](#_Toc121054001)

[Implementation (Hoàn thành) 19](#_Toc121054002)

[Storage Nodes (CÁC NÚT LƯU TRỮ) 19](#_Toc121054003)

[Client Driver ( TRÌNH ĐIỀU KHIỂN) 22](#_Toc121054004)

[Security (BẢO MẬT ) 23](#_Toc121054005)

[Integration (HỘI NHẬP) 24](#_Toc121054006)

[Oracle Database Integration (Tích hợp cơ sở dữ liệu Oracle) 24](#_Toc121054007)

[Hadoop, Hive and Big Data SQL Integration (Tích hợp SQL Hadoop, Hive và Dữ liệu lớn ) 25](#_Toc121054008)

[Performance ( HIỆU SUẤT) 26](#_Toc121054009)

[Conclusion ( kết luận) 27](#_Toc121054010)

# Giới thiệu

Cơ sở dữ liệu Oracle NoSQL cho phép tạo ra các ứng dụng sáng tạo thu hút khách hàng và tạo ra giá trị kinh doanh cho các yêu cầu kinh doanh liên tục thay đổi. Tổ chức của bạn có thể nhanh chóng đáp ứng các cơ hội mới yêu cầu lưu trữ và truy xuất dữ liệu rất nhanh với độ trễ thấp. Oracle NoSQL Database là một cơ sở dữ liệu NoSQL phân mảnh, có thể mở rộng, được thiết kế để cung cấp quản lý dữ liệu sẵn có, linh hoạt và có độ tin cậy cao  trên một tập hợp các nút lưu trữ có thể định cấu hình.

I Cơ sở dữ liệu Oracle NoSQL đã được thiết kế linh hoạt trong một số lĩnh vực:

\* Các nhà phát triển có thể tạo các ứng dụng đổi mới bằng một số ngôn ngữ lập trình phổ biến và có thể mô hình hóa dữ liệu theo một số cách.

\* Quản trị viên có thể lập kế hoạch cho các số lượng công việc khác nhau bằng cách thay đến quy mô theo cả chiêu dọc và chiều ngang, sử dụng máy chủ tiêu chuẩn trong ngành

## Tổng quan về Oracle NoSQL Database

  Cơ sở dữ liệu noSQL đại diện cho sự phát triển gần đây trong Mô hình ứng dụng doanh nghiệp, tiếp tục quá trình phát triển của 20 năm qua. Vào những năm 1990, các ứng dụng được tích hợp theo chiều dọc đã nhường cho Mô hình máy khách-máy chủ và gần đây hơn, Mô hình máy khách-máy chủ nhường chỗ cho ứng dụng web 3 tầng. Đồng thời , nhu cầu về các dịch vụ quy mô web đã bổ sung khả năng truy cập thời gian có độ trễ thấp cũng như xử lý thu nhỏ bản đồ ngoại tuyến. Mô hình dữ liệu bắt đầu tránh tính nhất quán của giao dịch để đổi lấy khả năng mở rộng gia tăng và phân phối quy mô lớn. Phong trào NoSQL nổi lên từ hệ sinh thái thứ hai này.

   NoSQL thường đặc trưng bởi  nó không - tùy thuộc vào người bạn hỏi, nó không chỉ là hệ thống quản lý cơ sở dữ liệu quan hệ dựa trên SQL, hoặc nó đơn giản không phải là Hệ thống quản lý cơ sở dữ liệu quan hệ dựa trên SQL (RDBMS), trong khi các định nghĩa đó giải thích NoSQL là  không, họ không giải thích được NoSQL là gì. Hãy xem xét các nguyên tắc cơ bản đã hướng dẫn quản lý dữ liệu trong bốn mươi năm qua. Các hệ thống RDBMS và quản lý dữ liệu quy mô lớn được đặc trưng bởi các thuộc tính ACID giao dịch của Tính nguyên tử, Tính nhất quán, tính cô lập và Độ bền. Ngược lại, NoSQL đôi khi được đặc trưng bởi từ viết tắt BASE:

   Về cơ bản: Sử dụng tính năng sao chép để giảm khả năng dữ liệu không có sẵn và sử dụng phân đoạn dữ liệu (phân vùng dữ liệu giữa nhiều máy chủ lưu trữ khác nhau) để loại bỏ bất kỳ lỗi nào còn lại. Với sự ra đời của sharding và nhân rộng để sẵn sàng mang lại lợi ích của kiến ​​trúc bảo mật và mở rộng quy mô theo chiều ngang.  Kết quả là một hệ thống có khả năng mở rộng quy mô lớn và luôn sẵn sàng, ngay cả khi các tập hợp con của dữ liệu trở nên không khả dụng trong một khoảng thời gian ngắn.

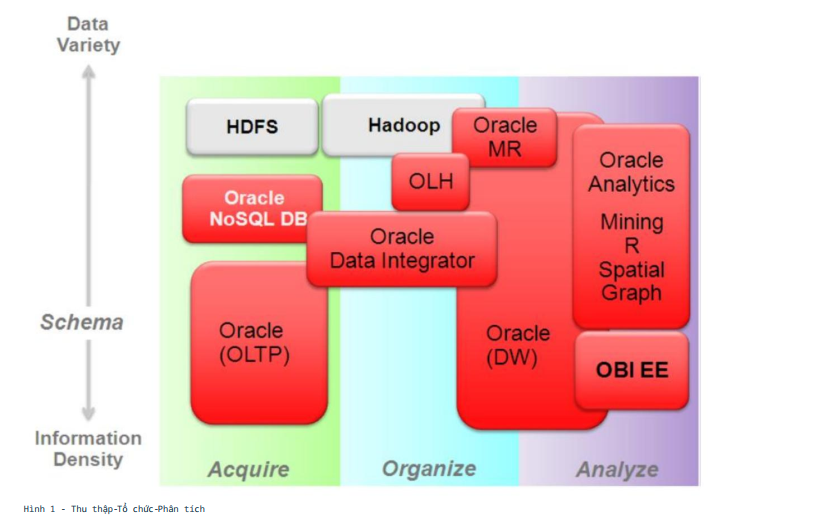
Trạng thái mềm: Trong khi các hệ thống ACID cho rằng tính nhất quán của dữ liệu là một yêu cầu khó, cơ sở dữ liệu NoSQL cho phép dữ liệu không nhất quán và hủy bỏ thiết kế xung quanh sự không nhất quán như vậy đối với các nhà phát triển ứng dụng.

Cuối cùng nhất : Mặc dù các ứng dụng phải đối phó với sự không nhất quán tức thời, các hệ thống NoSQL đảm bảo rằng tại một số thời điểm trong tương lai, dữ liệu sẽ có trạng thái nhất quán. Trái ngược lại các hệ thống ACID thực thi tính nhất quán khi cam kết giao dịch, NoSQL chỉ đảm bảo tính nhất quán tại một số thời điểm không xác định trong tương lai.

NoSQL xuất hiện khi các công ty như Amazon, Google, LinkedIn và Yahoo phải vật lộn để xử lý một lượng dữ liệu chưa từng có và khối lượng hoạt động trong điều kiện hạn chế chặt chẽ về đồ trễ. việc Phản hồi trong thời gian thực hoặc gần thời gian thực cho hàng triệu yêu cầu trong khi duy trì độ trễ có thể dự đoán được đã trở thành trình điều khiển kỹ thuật cốt lõi để mang lại trải nghiệm  phong phú cho người dùng , vị trí hệ thống đặt quảng cáo nhắm tới  mục tiêu cao và thu thập khối lượng lớn dữ liệu do máy tạo để phân tích ngoại tuyến tiếp theo. Cơ sở dữ liệu quan hệ truyền thống không đáp ứng được nhiệm vụ, vì vậy các doanh nghiệp đã dành trên một thập kỉ để xây dựng nghiên cứu về bảng hàm phân phối (DHT) và hệ thống cơ sở dữ liệu quan hệ độc đáo hoặc kho lưu trữ khóa/giá trị nhúng, chẳng hạn như Cơ sở dữ liệu Berkeley của Oracle (BDB), để phát triển các cửa hàng khóa-giá trị phân tán, có tính sẵn sàng cao.

Mặc dù nhiều cơ sở dữ liệu NoSQL đã phát triển trong vài năm qua, nhưng rất ít cơ sở dữ liệu có thể cung cấp các khái niệm về mô hình giao dịch cách linh hoạt. Thông thường, cơ sở dữ liệu NoSQL phải đánh đổi khả năng cung cấp cho nhà phát triển chế độ xem dữ liệu nhất quán và riêng biệt để lấy hiệu suất, quy mô, tính dễ lập trình hoặc tất cả những điều trên. Ngược lại, Cơ sở dữ liệu Oracle NoSQL cung cấp một mô hình ACID/ BASE kết hợp với tác động đến hiệu suất và quy mô gần như bằng không trong khi đạt được sự dễ dàng trong lập trình khiến các nhà phát triển hài lòng. các Mô hình sư ứng dụng và các nhà phát triển có thể quyết định thời điểm thích hợp để nới lỏng các ràng buộc giao dịch hoặc thắt chặt chúng. Với mỗi lần truy cập vào cơ sở dữ liệu, mức độ nhất quán thích hợp (đối với lần đọc) và độ bền (đối với lần ghi) có thể được chọn dựa trên nhu cầu cụ thể của ứng dụng.

Các tổ chức đã sử dụng công nghệ NoSQL như một khía cạnh công nghệ trong Mô hình ứng dụng doanh nghiệp và Cơ sở dữ liệu NoSQL của Oracle cung cấp tất cả các tính năng mong muốn của các giải pháp NoSQL cần thiết để tích hợp liền mạch vào bất kỳ Mô hình ứng dụng doanh nghiệp nào. Hình 1 cho thấy một chu kỳ tổ chức-thu thập-phân tích chuẩn, thể hiện cách Cơ sở dữ liệu NoSQL của Oracle phù hợp với một hệ sinh thái như vậy. Các mô-đun do Oracle cung cấp cho phép Cơ sở dữ liệu Oracle NoSQL tích hợp với Hadoop MapReduce, Hive, Spark hoặc với Cơ sở dữ liệu Oracle. Việc sử dụng Cơ sở dữ liệu Oracle để truy cập dữ liệu Cơ sở dữ liệu Oracle NoSQL giúp nâng cao bối cảnh tích hợp cho doanh nghiệp, cho phép công cụ BI hiện có, Oracle Enterprise R hoặc Phân tích nâng cao của Oracle truy cập dữ liệu được lưu trữ trong Cơ sở dữ liệu Oracle NoSQL.



HÌNH 1

Cơ sở dữ liệu Oracle NoSQL, với Mô hình “Không có 1 điểm lỗi nào” là giải pháp phù hợp khi việc truy cập dữ liệu về bản chất là “đơn giản” và nhu cầu ứng dụng vượt quá khả năng về khối lượng hoặc độ trễ của các giải pháp quản lý dữ liệu truyền thống. Ví dụ: dữ liệu luồng nhấp chuột từ các trang web có khối lượng lớn, xử lý sự kiện thông lượng cao và dữ liệu cảm biến internet vạn vật (IoT) đều đại diện cho các miền ứng dụng tạo ra khối lượng lớn dữ liệu có khóa đơn giản. Theo dõi hành vi bán lẻ trực tuyến, truy cập hồ sơ khách hàng, lấy quảng cáo khách hàng phù hợp, lưu trữ và chuyển tiếp thông tin liên lạc theo thời gian thực là những ví dụ về các miền yêu cầu tối ưu trong truy cập độ trễ thấp. Các ứng dụng phân tán cao như tổng hợp cảm biến thời gian thực và xác thực có thể mở rộng cũng đại diện cho các miền rất phù hợp với Cơ sở dữ liệu Oracle NoSQL.

# TỔNG QUAN VỀ KỸ THUẬT:

Cơ sở dữ liệu Oracle NoSQL tận dụng công cụ lưu trữ, tính khả dụng cao Phiên bản Java của Cơ sở dữ liệu Oracle Berkeley Java Edition để cung cấp khóa/ giá trị và bảng lưu trữ phân tán, có tính sẵn sàng cao cho các ứng dụng hoặc dịch vụ web có dung lượng lớn, nhạy cảm với độ trễ. Trong khi Oracle Berkeley DB cung cấp chức năng quản lý lưu trữ cơ bản, các lớp trên của Cơ sở dữ liệu Oracle NoSQL cung cấp các tính năng quan trọng như mở rộng đàn hồi trực tuyến, hỗ trợ nhiều mô hình dữ liệu (khóa/giá trị, bảng, JSON ((JavaScript Object Notation) tài liệu và đồ thị), Truy vấn SQL, tìm kiếm toàn văn, xác thực, ủy quyền và khắc phục thảm họa đa trung tâm dữ liệu.

## Architectural Data Model (Mô hình dữ liệu Mô hình):

Về cốt lõi, Cơ sở dữ liệu Oracle NoSQL triển khai bản đồ khóa/giá trị từ các khóa do người dùng xác định đến các mục dữ liệu không rõ ràng. Nó ghi lại số phiên bản cho các cặp khóa/dữ liệu nhưng duy trì một phiên bản mới nhất trong cửa hàng. Các ứng dụng không cần quan tâm đến việc điều chỉnh các phiên bản không tương thích vì Cơ sở dữ liệu Oracle NoSQL sử dụng bản sao dựa trên trình dẫn đầu; Paxos luôn có giá trị cập nhật nhất cho một khóa nhất định, trong khi các bản sao không chính chủ có thể có các phiên bản cũ hơn một chút. Các ứng dụng có thể sử dụng số phiên bản để đảm bảo tính nhất quán cho các thao tác đọc-sửa đổi ghi.

## Data Modeling Options (Tùy chọn mô hình hóa dữ liệu):

Là một kho lưu trữ dữ liệu đa mô hình thực sự, Cơ sở dữ liệu Oracle NoSQL cung cấp một số tùy chọn khác nhau để lập mô hình dữ liệu:

-Các cặp khóa/giá trị – Sử dụng tùy chọn lập mô hình này, nhà phát triển chỉ định các khóa chuỗi và mảng byte mờ làm giá trị. Cơ sở dữ liệu Oracle NoSQL không diễn giải các mảng byte. Chúng được ứng dụng tuần tự hóa và hủy tuần tự hóa.

-TABLE- Sử dụng tùy chọn mô hình hóa này, các nhà phát triển chỉ định các định nghĩa bảng bằng ngôn ngữ định nghĩa dữ liệu SQL, rất giống với cơ sở dữ liệu quan hệ. Dòng lệnh SQL để truy vấn Cơ sở dữ liệu Oracle NoSQL hoặc Cơ sở dữ liệu Oracle NoSQL  APIs có thể được sử dụng để truy cập dữ liệu bảng.

-Đồ thị – Sử dụng tùy chọn lập mô hình này, nhà phát triển xác định bất kỳ số lượng cặp tên/giá trị nào cho các cạnh hoặc đỉnh của biểu đồ. Sau đó, các cạnh được tạo để vẽ các mối quan hệ “được gắn nhãn” giữa các đỉnh của đồ thị.

-JSON documents – Sử dụng tùy chọn lập mô hình này, nhà phát triển có thể lưu trữ các đối tượng JSON trong Oracle NoSQL. Cơ sở dữ liệu Oracle NoSQL cung cấp một phương ngữ SQL phong phú với các toán tử chuyên biệt được thiết kế đặc biệt để cắt qua các đối tượng JSON. Các chỉ mục phụ được chỉ định bởi các biểu thức đường dẫn JSON cũng được hỗ trợ thông qua phương ngữ SQL này.

Nhà phát triển có thể chọn sử dụng cặp khóa/giá trị thô hoặc bảng để lập mô hình dữ liệu cho ứng dụng của họ.

🡺 Một số yếu tố cần được xem xét khi chọn mô hình hóa dữ liệu theo cách này hay cách khác:

-Chỉ số phụ hoặc chỉ số văn bản cần thiết? Nếu ứng dụng của bạn có thể hưởng lợi từ chỉ mục phụ hoặc chỉ mục toàn văn bản, thì bạn phải sử dụng bảng thay vì cặp khóa/giá trị thô để lập mô hình dữ liệu của mình. Cơ sở dữ liệu Oracle NoSQL sử dụng định nghĩa bảng để cho phép tạo chỉ mục phụ.

-Cần ủy quyền chi tiết? Nếu bạn cần bảo mật dữ liệu của mình bằng cách sử dụng ủy quyền chi tiết, phải sử dụng bảng thay vì cặp khóa/giá trị thô để lập mô hình dữ liệu của bạn.

- Truy vấn SQL – Bạn  định truy cập dữ liệu của mình bằng lệnh gọi SQL hay APIs? Nếu bạn có kế hoạch  sử dụng SQL, thì bạn nên sử dụng bảng thay vì cặp khóa/giá trị thô để lập mô hình dữ liệu của bạn.

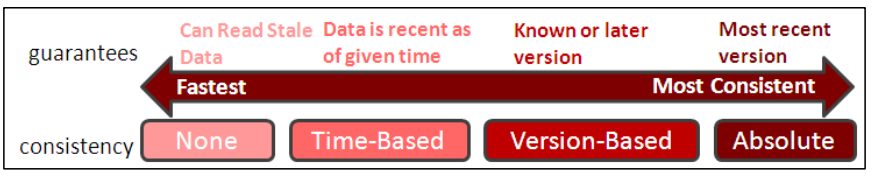
-Tài liệu JSON – Ứng dụng của bạn có được thiết kế để sử dụng JSON làm định dạng dữ liệu chính không? Nếu JSON này cần được lưu trữ liên tục và được truy vấn, thì bạn có thể sử dụng mô hình bảng và kiểu dữ liệu JSON đã được liên kết.

## Sharding Model (MÔ HÌNH CHIA NHỎ)

Cơ sở dữ liệu Oracle NoSQL chia nhỏ các khóa thành các phân đoạn để cung cấp phân phối trên một tập hợp các nút lưu trữ cung cấp lưu trữ cho cơ sở dữ liệu. Tuy nhiên, các ứng dụng có thể tận dụng khả năng của khóa con (còn được gọi là bảng con) để đạt được vị trí dữ liệu. Khóa chính là khả năng ghép nối của khóa phân đoạn và khóa không phải phân đoạn, cả hai đều được chỉ định bởi ứng dụng. Tất cả các bản ghi chia sẻ khóa phân đoạn được đặt cùng vị trí để đạt được dữ liệu địa phương. Trong một bộ sưu tập đồng vị của các khóa mảnh vỡ, khóa chính đầy đủ, bao gồm cả phần phân mảnh và phần không phân mảnh, cung cấp tra cứu được lập chỉ mục nhanh chóng.. Ví dụ: một ứng dụng lưu trữ email của người dùng có thể sử dụng ID người dùng làm khóa phân đoạn và sau đó có một số bảng con cho các thư mục email khác nhau do người dùng sở hữu, chẳng hạn như hộp thư đến, đã xóa, bản nháp, v.v. Khi hiển thị giao diện người dùng khi đăng nhập, trang đầu tiên trong hộp thư đến của người dùng (hoặc bất kỳ thư mục nào khác cho người dùng đó) có thể được đọc bằng một lệnh gọi APIs duy nhất từ ứng dụng. Nói cách khác, một cuộc gọi mạng duy nhất có thể được thực hiện tới Cơ sở dữ liệu Oracle NoSQL để truy xuất tất cả các thông báo cho một người dùng cụ thể.

## Consistency and Durability Model (Tính nhất quán và độ bền) :

Trong khi nhiều cơ sở dữ liệu NoSQL cung cấp tính nhất quán cuối cùng, thì Cơ sở dữ liệu Oracle NoSQL cung cấp một số chính sách nhất quán khác nhau. Ở một đầu của quang phổ, các ứng dụng có thể chỉ định tính nhất quán tuyệt đối, điều này đảm bảo rằng tất cả các lần đọc trả về giá trị được ghi gần đây nhất cho một khóa được chỉ định. Mặt khác, các ứng dụng có khả năng chịu đựng dữ liệu không nhất quán có thể chỉ định tính nhất quán yếu, cho phép cơ sở dữ liệu trả về một giá trị hiệu quả ngay cả khi nó không hoàn toàn cập nhật. Ở giữa hai thái cực này, các ứng dụng có thể chỉ định tính nhất quán dựa trên thời gian để hạn chế độ cũ của bản ghi đối với phiên bản mới nhất ở đầu hoặc tính nhất quán dựa trên phiên bản để hỗ trợ cả tính nguyên tử cho các hoạt động đọc-sửa-ghi và các lần đọc ở ít nhất là gần đây nhất như phiên bản được chỉ định. Hình 2 cho thấy phạm vi chính sách nhất quán linh hoạt cho phép nhà phát triển dễ dàng tạo giải pháp kinh doanh cung cấp đảm bảo dữ liệu trong khi vẫn đáp ứng các yêu cầu về độ trễ và khả năng mở rộng của ứng dụng.



HÌNH 2 : CAP Model

Cơ sở dữ liệu Oracle NoSQL cũng cung cấp một loạt các chính sách về độ bền xác định những gì đảm bảo hệ thống sẽ thực hiện sau sự cố. Ở một bên khác, các ứng dụng có thể yêu cầu khối yêu cầu ghi đó cho đến khi bản ghi được ghi vào bộ lưu trữ ổn định trên tất cả các bản sao. Điều này có ý nghĩa rõ ràng về hiệu suất và tính khả dụng nhưng đảm bảo rằng nếu ứng dụng ghi dữ liệu thành công, dữ liệu đó sẽ tồn tại và có thể được khôi phục ngay cả khi tất cả các bản sao tạm thời không khả dụng do nhiều lỗi đồng thời. Ở một bên khác, các ứng dụng có thể yêu cầu các hoạt động ghi trở lại ngay sau khi hệ thống đã ghi lại sự tồn tại của hoạt động ghi, ngay cả khi dữ liệu không tồn tại ở bất kỳ đâu. Chính sách như vậy cung cấp hiệu suất ghi tốt nhất nhưng không đảm bảo độ bền. Bằng cách chỉ định thời điểm cơ sở dữ liệu ghi các bản ghi vào đĩa và phần nào trong số các bản sao của bản ghi phải được lưu giữ lâu dài (không có, tất cả hoặc đa số đơn giản), các ứng dụng có thể thực thi một loạt các chính sách về độ bền và tạo ra sự đánh đổi quan trọng giữa độ trễ và độ bền.

## Programming Models (Mô hình lập trình):

Như đã nêu trước đây, các nhà phát triển có nhiều lựa chọn về cấu trúc mô hình hóa trong Cơ sở dữ liệu Oracle NoSQL và những cấu trúc mô hình hóa dữ liệu này dẫn trực tiếp đến mô hình lập trình. Cơ sở dữ liệu Oracle NoSQL cho phép các nhà phát triển vừa lập mô hình dữ liệu của họ vừa sử dụng một APIs tự nhiên cùng nhau.

Ví dụ: nếu bạn cần sử dụng JSON làm mô hình dữ liệu của mình, thì APIs cơ sở dữ liệu Oracle NoSQL JSON đã được phát triển để đơn giản hóa việc phát triển ứng dụng của bạn. Tương tự như vậy, việc sử dụng các bảng để lập mô hình ứng dụng của bạn sẽ dẫn đến việc sử dụng tự nhiên APIS Cơ sở dữ liệu của Oracle NoSQL. Trong bài báo này, chúng tôi sẽ đưa ra các ví dụ về hai mô hình cụ thể này.

## The Table Model (Mô hình bảng) :

Một trong những mô hình dữ liệu được Cơ sở dữ liệu Oracle NoSQL đưa ra là mô hình bảng. Mặc dù mô hình này có thể quen thuộc hơn với các nhà phát triển có kinh nghiệm với cơ sở dữ liệu quan hệ, nhưng mô hình bảng Cơ sở dữ liệu Oracle NoSQL cung cấp các kiểu dữ liệu duy nhất không tìm thấy trong thế giới quan hệ, chẳng hạn như mảng, bản đồ và bản ghi nhúng. Việc tạo các bảng trong Cơ sở dữ liệu Oracle NoSQL được thực hiện thông qua Ngôn ngữ Định nghĩa Dữ liệu SQL (DDL), rất giống với SQL quan hệ. Dưới đây là một câu lệnh DDL  sẽ tạo một bảng có tên Người dùng với các cột sau:

> Id- Đây là khóa chính cũng như là khóa phân đoạn (trừ khi được chỉ định, còn bình thường khóa phân đoạn sẽ mặc định là khóa chính)

> FirstName and lastName là chuổi để lưu trữ họ và tên người dùng

>> Mảng các bản ghi địa chỉ- Chúng sẽ lưu trữ nhiều địa chỉ cho người dùng. Mỗi bản ghi địa chỉ sẽ chứa các thuộc tính sau:

  >> Street,City,State -Đường, thành phố và các tiểu bang

>> Zip - Mã số nguyên địa chỉ này

  >addrType -Một phép liệt kê có thể chứa “ cơ quan” hoặc “nhà riêng”

Trong Cơ sở dữ liệu Oracle NoSQL, DDL để tạo bản này có thể được chỉ định là :

*Kết quả khai báo res = KVStore.executeSync(“CREATE TABLE user(id INTEGER,*

*firstName STRING, lastName String,*

*addresses ARRAY(RECORD (street STRING, city String, state STRING,*

*zip INTEGER, addr Type ENUM(nhà riêng, cơ quan),*

*primary key(id))))”);*

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

## Accessing Table Data via SQL (Truy cập bảng dữ liệu qua SQL) :

Đối với các truy vấn không phù hợp với kiểu truy cập khóa/giá trị đơn giản hoặc quét pham 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  cũng cấp một phương ngữ SQL khá phong phú, nhưng SQL này có thể được coi là một tập hợp con đơn giản hơn của SQL mạnh mẽ và toàn diện hơn do Oracle RDBMS cung cấp. Đối với những ứng dụng yêu cầu mức truy cập SQL này thì cơ sở dữ liệu Oracle NoSQL được tích hợp với Oracle RDBMS, cho phép RDBMS truy vẫn dữ liệu trực tiếp từ các bảng cơ sở dữ liệu Oracle NoSQL.

Một lĩnh vực khác biệt đáng chú ý giữa các phương ngữ Cơ sở dữ liệu Oracle RDBMS và Oracle NoSQL của SQL là khái niệm truy vấn SQL-ability so với dữ liệu không có quan hệ như mảng và bản đồ.. Phương ngữ 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.

1. Trả lại tên họ tất cả người dùng của thành phố New York

*select*

*firstName,*

*lastName,*

*from*

*user*

*where*

*addresses.phones[$element.city = “New York”]*

1. Trả lại họ và tên của tất cả người dùng có địa chỉ nhà trong mã zip 94107 và đặt kết quả của thuộc tính lastName:

*select*

*firstName,*

*lastName*

*from*

*user*

*where*

*addresses.phones[$element.zip= 94107] and*

*addresses.phones[$element.attrType = “home”] order 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:

*select \**

*from*

*user*

*where*

*addresses.phones[$element.city = “New York”] or*

*addresses.phones[$element.city = “New Haven”] or*

*addresses.phones[$element.city = “Newark”]*

*order by lastName limit 25 offset 1*

## Working with JSON Documents ( 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 lệnh gọi API dịch vụ và trao đổi dữ liệu giữa các ứng dụng. Là một loại dữ liệu tự mô tả, dữ liệu được định dạng JSON cung cấp tính linh hoạt mớimới 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 có 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 có lược đồ và phần nào của ứng dụng sẽ tối ưu hóa việc lưu trữ và tính toán để đổi lấy một lược đồ cố định.

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

*Kết quả câu lệnh 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 phương ngữ SQL hoạt động trên Bảng. Ngoài ra, đối với các tài liệu JSON đặc biệt, một số toán tử mới và biểu thức "case" được giới thiệu để hỗ trợ xử lý các tài liệu có thể có định dạng rời rạc hoặc không xác định. 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 lệnh gọi API dịch vụ và trao đổi dữ liệu giữa các ứng dụng. Là một loại 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 riêng nó.

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 thì áp dụng vị ngữ trực tiếp cho một thuộc tính có tên là “city” trong tài liệu chính.

*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)*

2. Tìm tất cả người dùng ở Thành phố New York khi tài liệu JSON chứa thuộc tính thành phố, trong tài liệu cho lược đồ. vị từ 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. 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 API hoặc truy vấn SQL từ một hoặc nhiều quy trình máy khách truy cập máy chủ Cơ sở dữ liệu Oracle NoSQL độc lập quy trình, ggiảm bớt sự cần thiết phải thiết lập các cấu hình đa hệ thống để phát triển và thử nghiệm ban đầu.

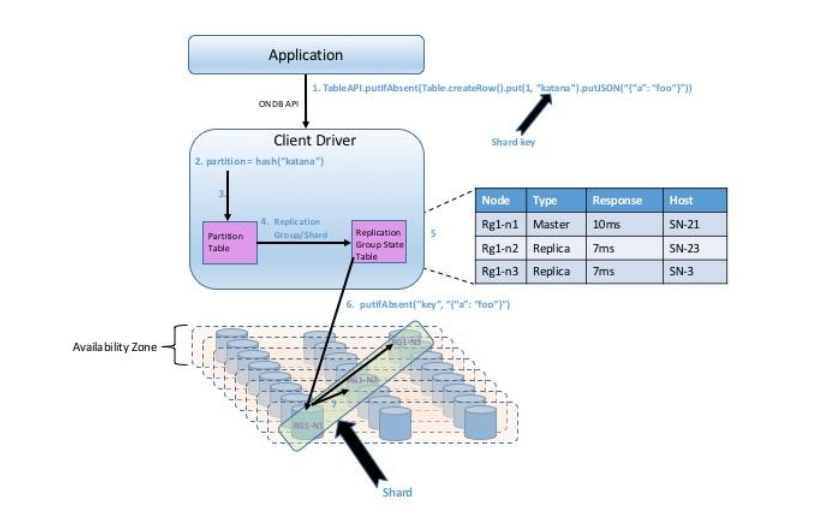
# Failure Resiliency (khả năng chống lỗi)

Cơ sở dữ liệu Oracle NoSQL đưa ra khái niệm về vùng khả dụng như các vùng chứa cách ly lỗi. Vùng khả dụng có thể có dạng một giá đỡ các máy chủ, một đầu chuyển mạch giá đỡ, tầng của một tòa nhà hoặc toàn bộ trung tâm dữ liệu. Các khu cũng được đặc trưng là chính hoặc phụ, có các khu chính dành cho các cuộc bầu cử tổng thể cũng như xác nhận số giao thức. Cơ sở dữ liệu Oracle NoSQL tiếp tục cho phép cấu hình bất kỳ số lượng vùng chính và vùng phụ nào. Sau khi được định cấu hình, các bản sao cho mỗi phân đoạn trong kho dữ liệu được bố trí trên các vùng này sao cho việc hỏng bất kỳ vùng nào sẽ hạn chế (hoặc chứa) lỗi này để số giao thức luôn được duy trì bởi mọi mỗi mảnh vỡ trong lõi dữ liệu

Cơ sở dữ liệu Oracle NoSQL triển khai một Dịch vụ quản trị riêng có tính khả dụng cao. phù hợp với cơ sở dữ liệu Oracle NoSQL với triết lý “Không có điểm lỗi”, hoạt động liên tục của quá trình cài đặt không phụ thuộc vào sự sẵn có của Dịch vụ quản trị. Như vậy, cả cơ sở dữ liệu và quản trị dịch vụ vẫn khả dụng trong quá trình thay đổi cấu hình.

# Architecture (Mô hình)

Chúng tôi trình bày mô hình 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 phần cứng và phần mềm thực tế hoạt động. 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”}



HÌNH 3: Mô hình- CƠ SỞ DỮ LIỆU ORACLE NOSQL- LUỒNG YÊU CẦU GHI

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

Một phân đoạn bao gồm một số nút sao chép (có thể định cấu hình) 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 cho biết số lượng sự cố mà 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 yêu cầu đọc. Khả năng chịu được lỗi khi ghi của nó dựa trên độ bền được yêu cầu chính sách trong lệnh gọi API ghi. Nếu ứng dụng không yêu cầu đa số người tham gia xác nhận một bài viết, thì hệ thống cũng có thể chịu được tối đa hai lỗi ghi. Một nhóm năm nút có thể chịu được tối đa bốn lỗi đối với đọc và tối đa hai lỗi ghi, ngay cả khi ứng dụng yêu cầu chính sách độ bền yêu cầu phần lớn các trang web để xác nhận một 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 mảnh,RGST chứa thông tin về mỗi nút sao chép bao gồm nhóm các nút sao chép trong mảnh vỡ (bước 5). Dựa trên thông tin trong RGST, chẳng hạn như danh tính của chủ và tải trên các nút khác nhau trong một 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).

# Im**plementation (Hoàn thành)**

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 các thành phần.

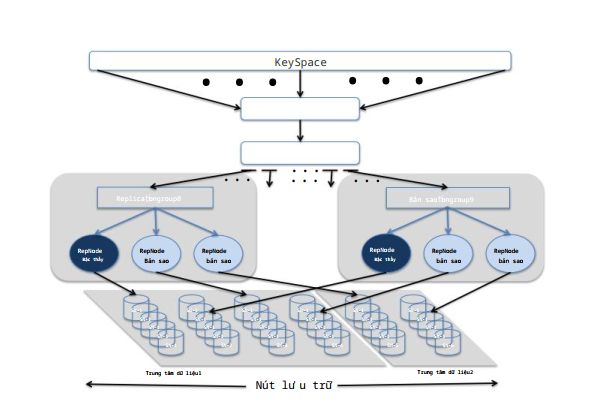
## Storage Nodes (các nút lưu trữ)

A storage node (SN) thường là một máy vật lý có bộ lưu trữ liên tục cục bộ, dạng đĩa hoặc trạng solid state,1 CPU với 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.

 A Storage Node Agent  (SNA) chạy trên mỗi nút lưu trữ, theo dõi hành vi của nút đó. SNA vừa nhận cấu hình vừa báo cáo thông tin giám sát cho Dịch vụ quản trị. SNA liên tục thu thập dữ liệu vận hành từ nút lưu trữ và sau đó cung cấp dữ liệu đó cho Dịch vụ quản trị khi được yêu cầu.

Một nút lưu trữ phục vụ một hoặc nhiều nút sao chép. Mỗi nút sao chép thuộc về một nhóm sao chép duy nhất. Các nút trong một nhóm sao chép duy nhất đều phục vụ cùng một dữ liệu. Mỗi nhóm có một nút chính được chọn động để xử lý tất cả các hoạt động sửa đổi dữ liệu (tạo, cập nhật và xóa). Các nút khác là bản sao chỉ đọc nhưng có thể đảm nhận vai trò chính nếu nút chính bị lỗi. Một cài đặt điển hình sử dụng hệ số sao chép là ba trong các nhóm sao chép, để đảm bảo rằng hệ thống có thể tồn tại ít nhất hai lỗi đồng thời và tiếp tục phục vụ các hoạt động đọc. Các ứng dụng yêu cầu độ tin cậy cao hơn hoặc thấp hơn có thể điều chỉnh thông số này cho phù hợp

Hình 4 cho thấy một cài đặt với 10 nhóm sao chép (0-9) (còn được gọi là phân đoạn). Mỗi nhóm sao chép có hệ số sao chép là 3 (một bản chính và hai bản sao) trải rộng trên hai trung tâm dữ liệu. Lưu ý rằng chúng ta có thể đặt hai trong số các nút sao chép vào trung tâm dữ liệu lớn hơn và nút sao chép cuối cùng vào trung tâm dữ liệu nhỏ hơn. Kiểu sắp xếp này có thể phù hợp với ứng dụng sử dụng trung tâm dữ liệu lớn hơn để truy cập dữ liệu chính, duy trì trung tâm dữ liệu nhỏ hơn trong trường hợp trung tâm dữ liệu chính gặp sự cố nghiêm trọng



HÌNH 4: MÔ HÌNH SỬ DỤNG 10 SHARD VÀ RF=3

10 nhóm sao chép được lưu trữ trên 30 nút lưu trữ, trải rộng trên hai trung tâm dữ liệu.

Các nút sao chép hỗ trợ API Cơ sở dữ liệu Oracle NoSQL thông qua các lệnh gọi RMI từ máy khách và lấy dữ liệu trực tiếp hoặc ghi dữ liệu trực tiếp vào hệ thống lưu trữ có cấu trúc nhật ký, cung cấp khả năng ghi hiệu suất cao, đồng thời duy trì cấu trúc chỉ mục cung cấp hiệu suất đọc có độ trễ thấp.

Cơ sở dữ liệu Oracle NoSQL sử dụng bản sao để đảm bảo tính khả dụng của dữ liệu trong trường hợp lỗi. Mô hình một chủ của nó yêu cầu ghi được áp dụng tại nút chính và sau đó được truyền tới các bản sao. Trong trường hợp nút chính bị lỗi, các nút trong nhóm sao chép sẽ tự động tổ chức một cuộc bầu cử đáng tin cậy (sử dụng giao thức Paxos), chọn một trong các nút còn lại làm nút chính. Master mới sau đó đảm nhận trách nhiệm ghi.

## Client Driver ( TRÌNH ĐIỀU KHIỂN)

Trình điều khiển máy khách là một tệp jar Java xuất API sang các ứng dụng. Trình điều khiển máy khách nhận biết cấu trúc liên kết bằng cách sử dụng Bảng trạng thái nhóm nhân bản (RGST). Cấu trúc liên kết ánh xạ hiệu quả các khóa đến các phân vùng và từ các phân vùng đến các nhóm sao chép. Đối với mỗi nhóm sao chép, nó bao gồm tên máy chủ của nút lưu trữ lưu trữ mỗi nút sao chép trong nhóm, tên dịch vụ được liên kết với các nút sao chép và trung tâm dữ liệu mà mỗi nút lưu trữ nằm trong đó. Sau đó, khách hàng sử dụng RGST cho hai mục đích chính: xác định nút chính của một nhóm sao chép, để nó có thể gửi yêu cầu ghi tới máy chủ và cân bằng tải trên tất cả các nút trong nhóm sao chép để đọc. Vì RGST là một cấu trúc dữ liệu được chia sẻ quan trọng nên mỗi máy khách và nút sao chép sẽ duy trì bản sao của nó, do đó tránh được bất kỳ điểm lỗi đơn lẻ nào. Cả máy khách và nút sao chép đều chạy RequestDispatcher sử dụng RGST để (lại) hướng các yêu cầu ghi tới máy chủ và đọc các yêu cầu tới thành viên thích hợp của nhóm sao chép.

Cấu trúc liên kết được tải trong quá trình khởi tạo nút máy khách hoặc nút sao chép và sau đó có thể được quản trị viên cập nhật nếu có thay đổi cấu trúc liên kết. RGST là hoạt động, yêu cầu bảo trì liên tục. Mỗi nút sao chép chạy một luồng, được gọi là luồng Cập nhật trạng thái nút sao chép, chịu trách nhiệm duy trì liên tục RGST. Chuỗi cập nhật, cũng như RequestDispatcher, thu thập thông tin theo cơ hội trên các nút sao chép từ xa bao gồm trạng thái hiện tại của nút trong nhóm sao chép của nó, chỉ báo về mức độ cập nhật của nút, thời gian tương tác thành công cuối cùng với nút, thời gian phản hồi trung bình sau của nút và độ dài hiện tại của hàng đợi yêu cầu chưa xử lý của nút đó. Ngoài ra, luồng cập nhật duy trì các kết nối mạng và thiết lập lại các kết nối bị hỏng. Việc bảo trì này được thực hiện bên ngoài chu kỳ yêu cầu hoặc phản hồi của RequestDispatcher để giảm thiểu tác động của các kết nối bị hỏng đối với độ trễ.

# Security (bảo mật )

Cơ sở dữ liệu Oracle NoSQL có được cấu hình an toàn. Trong cấu hình an toàn, giao tiếp mạng giữa các máy khách NoSQL, tiện ích và các thành phần máy chủ NoSQL được mã hóa bằng SSL / TLS và tất cả các quy trình phải được tự xác thực với các thành phần mà chúng kết nối.

Có hai cấp độ bảo mật cần lưu ý. Đây là bảo mật mạng, cung cấp lớp bảo vệ bên ngoài ở cấp độ mạng và xác thực/ủy quyền người dùng. Bảo mật mạng được định cấu hình ở cấp hệ thống tệp thường trong quá trình cài đặt, trong khi xác thực / ủy quyền người dùng được quản lý thông qua các tiện ích NoSQL.

Cơ sở dữ liệu Oracle NoSQL cung cấp các khả năng sau để bảo mật kho lưu trữ và thiết lập độ phức tạp của mật khẩu:

  -Tiện ích cấu hình bảo mật. Cho phép cấu hình để thêm bảo mật cho Oracle NoSQL mới hoặc hiện có cài đặt cơ sở dữ liệu

-Phương thức xác thực. Cơ sở dữ liệu Oracle NoSQL cung cấp xác thực mật khẩu cho người dùng và hệ thống. Các phiên bản Enterprise Edition của Cơ sở dữ liệu Oracle NoSQL cũng hỗ trợ xác thực Kerberos.

- Dữ liệu được mã hóa trên mạng để ngăn chặn truy cập trái phép vào dữ liệu đó.

- Lưu trữ mật khẩu bên ngoài. Cơ sở dữ liệu Oracle NoSQL cung cấp hai loại phương thức lưu trữ mật khẩu bên ngoài có thể thao tác (một loại dành cho các triển khai CE). Các loại là văn bản rõ ràng và sử dụng Oracle (có sẵn với Oracle NoSQL Database Enterprise Edition).

-Chính sách bảo mật: Cơ sở dữ liệu Oracle NoSQL cho phép thiết lập các hành vi để đảm bảo một môi trường an toàn

- Ủy quyền dựa trên vai trò: Cơ sở dữ liệu Oracle NoSQL cung cấp các vai trò hệ thống, đặc quyền và vai trò do người dùng xác định trước cho người dùng. Bạn có thể đặt các đặc quyền mong muốn cho người dùng bằng cách cấp vai trò.

# Integration (hội nhập)

Cơ sở dữ liệu Oracle NoSQL (phiên bản Enterprise Edition), khi được bàn giao, đã sẵn sàng để tích hợp với nhiều công nghệ Oracle khác nhau như Cơ sở dữ liệu Oracle, Trình quản lý doanh nghiệp Oracle, Oracle Coherence, Đồ thị và không gian dữ liệu lớn của Oracle, Dịch vụ dữ liệu nghỉ ngơi của Oracle và các công nghệ mã nguồn mở chẳng hạn như Hadoop, Hive và Spark.

## Oracle Database Integration (Tích hợp cơ sở dữ liệu Oracle)

Để  đọc dữ liệu từ Cơ sở dữ liệu Oracle NoSQL vào Bảng bên ngoài của Cơ sở dữ liệu Oracle, hãy xác định Bảng bên ngoài bằng một hoặc nhiều tệp vị trí; các tệp vị trí sẽ không thực sự chứa bất kỳ dữ liệu nào mà thay vào đó chứa thông tin cấu hình liên quan đến việc kết nối với Cơ sở dữ liệu Oracle và Cơ sở dữ liệu Oracle NoSQL, các hạn chế truy vấn và định dạng.

Việc chỉ định nhiều tệp vị trí trong định nghĩa Bảng bên ngoài cho phép mức độ song song trong quá trình đọc dữ liệu. Khi Bảng bên ngoài được xác định, hãy chạy tiện ích Publish để "xuất bản" thông tin về Bảng bên ngoài và cách truy cập dữ liệu trong Cơ sở dữ liệu Oracle NoSQL. Thông tin này được viết dưới dạng tài liệu XML thành từng của (các) tệp vị trí.

Khi quá trình xuất bản hoàn tất, SQL SELECT có thể được thực thi đối với Bảng bên ngoài. Điều này sẽ khiến tập lệnh /exttab/bin/ nosql\_stream được gọi, do đó khiến lớp Preproc được gọi. Các tiền xử lý :

  + Đọc cấu hình và thông tin truy cập Cơ sở dữ liệu NoSQL từ tệp vị trí,

  + Đọc dữ liệu từ Cơ sở dữ liệu NoSQL.

  +Định dạng nó bằng trình định dạng do người dùng xác định hoặc định dạng mặc định và viết nó thành stdout.

## Hadoop, Hive and Big Data SQL Integration (Tích hợp SQL Hadoop, Hive và Dữ liệu lớn )

Cơ sở dữ liệu Hadoop to Oracle NoSQL được hỗ trợ bằng cách chạy các công việc Hadoop MapReduce đối với dữ liệu được lưu trữ trong bảng Cơ sở dữ liệu Oracle NoSQL. Lớp InputFormat tùy chỉnh xác định phương pháp đọc và ghi dữ liệu đầu vào MapReduce. Ngoài ra còn có một triển khai tùy chỉnh của InputSplit xác định cách lấy danh sách "phân tách" logic cho công việc và triển khai tùy chỉnh của lớp RecordRead để thực hiện công việc định vị và trả về các cặp Khóa/Giá trị được sử dụng bởi MapReduce và SerDe.

Oracle Big Data SQL cho phép người dùng sử dụng ngôn ngữ SQL để quản lý và thao tác dữ liệu được lưu trữ ở một số vị trí khác nhau trong các cơ sở dữ liệu khác nhau. Oracle Big Data SQL đạt được điều này bằng cách trình bày dữ liệu Cơ sở dữ liệu Oracle NoSQL dưới dạng các bảng bên ngoài nhằm nâng cao cho Cơ sở dữ liệu Oracle. Điều này được thực hiện bằng cách ánh xạ ngôn ngữ bên ngoài để truy cập dữ liệu từ các nguồn này tới các cấu trúc bên trong Cơ sở dữ liệu Oracle.

Oracle Big Data SQL không chỉ cho phép dễ dàng tích hợp dữ liệu từ các nguồn Hadoop và NoSQL, Big Data SQL còn tận dụng các cơ chế lưu trữ cơ bản để mang lại hiệu suất tốt nhất có thể. Công nghệ kéo xuống vị trí ngôn ngữ của Big Data SQL cho phép các vị từ trong các truy vấn được đưa ra trong Cơ sở dữ liệu Oracle được thực thi bởi các hệ thống từ xa và được đẩy vào các định dạng tệp nhất định.

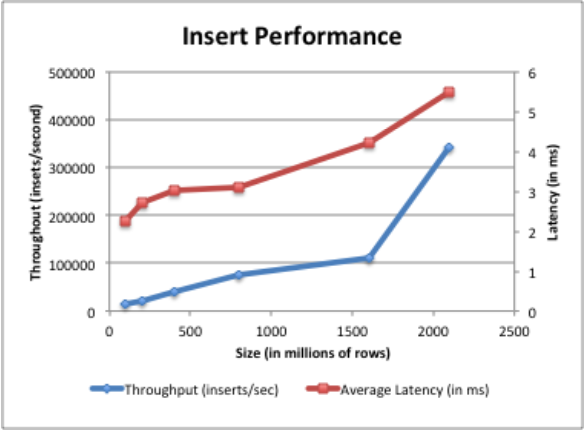
Tích hợp với Hive đạt được bằng cách chạy các truy vấn Hive bằng ngôn ngữ giống như SQL có tên là HiveQL (HQL) trên dữ liệu được lưu trữ trong Cơ sở dữ liệu NOSQL của Oracle. Hive TableStorageHandler tùy chỉnh xác định cơ chế được yêu cầu bởi cả Hive và SQL Dữ liệu lớn và được sử dụng khi thực hiện truy vấn đối với một bảng trong kho lưu trữ Cơ sở dữ liệu Oracle NoSQL. Ngoài ra, còn có lớp ObjectInspector tùy chỉnh được sử dụng để dịch các loại dữ liệu sang định dạng dữ liệu Cơ sở dữ liệu Oracle NoSQL tương ứng.

Để cải thiện hiệu suất truy vấn, cả Hive và Big Data SQL đều hỗ trợ một hình thức đẩy xuống trước trong đó frontend phía máy khách phân hủy trong ĐÓ điều khoản của một truy vấn thành thông tin cột và các hoạt động so sánh tương ứng.. Điều này dẫn đến việc đẩy các thành phần kết quả sang phía máy chủ cơ sở dữ liệu để xử lý. Big Data SQL sử dụng giao diện Hive để hỗ trợ đẩy xuống trước khi thực hiện truy vấn Big Data SQL đối với dữ liệu bảng KVStore.. Để hỗ trợ tính năng đẩy xuống , có một triển khai tùy chỉnh của HiveStoragePredicateHandler hỗ trợ và xác định tiêu chí phân tách các vị từ được đẩy xuống các  bảng và các  mục tiêu.

# Performance ( hiệu suất)

Chúng tôi đã thử nghiệm với nhiều cấu hình Cơ sở dữ liệu Oracle NoSQL khác nhau và trình bày một số kết quả về hiệu suất của Yahoo! Điểm chuẩn phục vụ đám mây (YCSB), thể hiện cách hệ thống mở rộng theo số lượng nút trong hệ thống. Như với tất cả các phép đo hiệu suất, kết quả có thể thay đổi do một số yếu tố.

Chúng tôi đã áp dụng tải YCSB không đổi trên mỗi nút lưu trữ cho các cấu hình có kích thước khác nhau. Mỗi nút lưu trữ bao gồm một cỗ máy ổ cắm kép Intel Xeon Processor X5670 với 6 lõi / ổ cắm và 24 GB bộ nhớ. Mỗi máy có một đĩa cục bộ duy nhất và chạy RedHat 2.6.18-164.11.1.el5.crt1. Tại 300 GB, kích thước đĩa là tài nguyên giới hạn tỷ lệ trên mỗi nút, quyết định cấu hình tổng thể, vì vậy chúng tôi đã cấu hình mỗi nút để giữ bản ghi 100M, với kích thước khóa trung bình là 13 byte và kích thước dữ liệu là 1108 byte.



HÌNH 5: KẾT QUẢ YSCB- HIỆU SUẤT KHI KÍCH THƯỚC DỮ LIỆU TĂNG LÊN

Hình 5 cho thấy hiệu suất chèn thô của Cơ sở dữ liệu Oracle NoSQL cho các cấu hình khác nhau hệ thống nhóm sao chép với ba nút lưu trữ 100 triệu bản ghi đến một hệ thống có 32 nhóm sao chép trên 96 nút lưu trữ 2,1 tỷ bản ghi (điểm chuẩn YCSB được giới hạn ở mức tối đa 2,1 tỷ bản ghi). Biểu đồ hiển thị cả thông lượng trong các hoạt động mỗi giây (đường màu xanh lam và trục bên trái) và thời gian phản hồi tính bằng mili giây (đường màu đỏ và trục bên phải). Thông lượng của hệ thống quy mô gần như tuyến tính khi kích thước cơ sở dữ liệu và số lượng nhóm sao chép tăng lên, với thời gian phản hồi chỉ tăng một chút.

# Conclusion ( kết luận)

Cơ sở dữ liệu NoSQL của Oracle mang lại chất lượng lưu trữ và hiệu suất cho doanh nghiệp. Môi trường NoSQL có tính khả dụng cao, được phân phối rộng rãi. Nó đã được chứng minh về mặt thương mại, hệ thống lưu trữ tối ưu hóa ghi mang lại hiệu suất vượt trội cũng như độ bền và độ tin cậy, đồng thời thiết kế “Không có điểm lỗi” của nó đảm bảo rằng hệ thống tiếp tục chạy và dữ liệu đó vẫn có sẵn sau một loạt khả năng xảy ra lỗi.