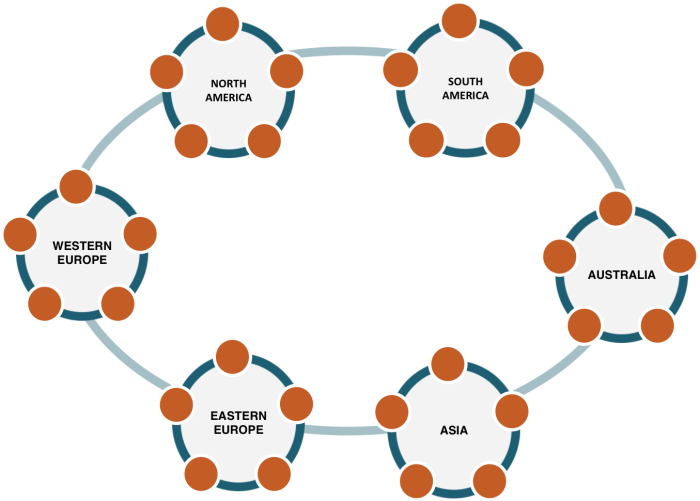
**TỔNG QUAN VỀ CASSANDRA**

1. **Cassandra là gì? (Giới thiệu tổng quát)**

* ***Cassandra***: Một quản trị hệ cơ sở dữ liệu phân tán mã nguồn mở, được thiết kế để xử lý một khối lượng lớn dữ liệu giàn trải trên nhiều node mà vẫn đảm bảo tính sẵn sàng cao (Highly Availability), khả năng mở rộng hay thu giảm số node linh hoạt (Elastic Scalability) và chấp nhận một số lỗi (Fault Tolerant).
* ***Thiết kế của Cassandra***: Thiết kế phân tán trên hàng ngàn máy chủ mà không có bất cứ điểm chết tập trung nào. Cassandra có thiết kế dựa trên kiến trúc mạng ngang hàng (Peer - to - Peer), tất cả các nút máy chủ trong hệ thống đều có vai trò như nhau và không có nút máy chủ nào đóng vai trò là máy chủ trung tâm.
* ***Các nút máy chủ***:
* Các nút máy chủ của Cassandra là độc lập và tham gia vào kết nối với các nút máy chủ khác trong hệ thống.
* Mỗi nút đều có thể xử lý các thao tác ghi và đọc dữ liệu, không phân biệt là dữ liệu được lưu trữ một cách vật lý trên máy chủ nào trong hệ thống.
* Khi một nút trong hệ thống bị hỏng và dừng hoạt động, các thao tác đọc ghi dữ liệu có thể được xử lý bởi các nút khác trong hệ thống => Quá trình này giúp ứng dụng không bị gián đoạn khi đang chạy và cho phép ẩn đi sự hư hỏng của nút trong hệ thống.
* ***Nhân bản dữ liệu***: Trong Cassandra, mỗi đối tượng dữ liệu có thể được nhân bản và lưu giữ trên nhiều máy chủ. Nếu một trong các máy chủ lưu một phiên bản dữ liệu bị lỗi hoặc không phải là phiên bản được cập nhật dữ liệu mới nhất, Cassandra có cơ chế đồng bộ để luôn đảm báo các thao tác đọc sẽ luôn trả về dữ liệu mới nhất. Cơ chế này được thực thi trong quá trình đọc dữ liệu (read repair) thay vì đồng bộ ngay trong thao tác ghi dữ liệu, điều này cho phép tăng hiệu năng cho thao tác ghi dữ liệu.
* ***Phân tán dữ liệu***: Cassandra sử dụng cơ chế hàm băm nhất quán phân tán (Distributed consistent hashing) tổ chức các nút máy chủ thành cụm theo định dạng vòng tròn và dữ liệu được phân tán theo vòng tròn này theo hàm băm nhất quán. Mỗi vòng tròn được coi là một Datacenter. Các nút trong một cụm Cassandra sẽ được phân bố trên một vòng tròn gọi là ring

*Hình ảnh minh họa*

* ***Model dữ liệu***:
* Các developer có thể không cần phải xác định trước model dữ liệu. Khi cần thêm cột, developer chỉ cần xác định tên cột tại thời điểm ghi dữ liệu và cột sẽ được tao ra nếu nó không tồn tại. Mỗi dòng (row) của cassandra có thể chứa một lượng rất lớn các cột (Cassandra hỗ trợ tới 2 tỷ column trên một row). Thêm vào đó số lượng các cột (column) trên mỗi dòng (row) có thể không giống nhau.
* Ngoài ra, Cassandra cung cấp sự lựa chọn cho việc tạo mẫu dữ liệu, nếu developer biết trước được mẫu dữ liệu, họ có thể tạo các column và chọn các kiểu dữ liệu cho column đó, đồng thời họ cũng có thể đánh index cho các cột dữ liệu này.

1. **Sự khác biệt của Cassandra (NoSQL) với những hệ quản trị cơ sở dữ liệu kiểu quan hệ (RDBMS)**

* ***RDBMS***
* Việc ánh xạ giữa các bảng trong database với các đối tượng trong code khá rắc rối và phức tạp. (*Mặc dù một số ORM như Entity Framework, Hibernate đã đơn giản hóa chuyện này)*.
* Performance sẽ bị chậm khi phải kết nhiều bảng để lấy dữ liệu (Đó là lý do ta sử dụng “giảm chuẩn” để tăng hiệu suất cho RDBMS).
* Việc thay đổi cấu trúc dữ liệu (Thêm/xóa bảng hoặc thêm/xóa một field) rất mệt mỏi, kéo theo vô số thay đổi trên code.
* Không làm việc được với dữ liệu không có cấu trúc (un-structure).
* RDBMS được thiết kế để chạy trên một máy chủ. Khi muốn mở rộng, nó khó chạy trên nhiều máy (clustering).
* ***NoSQL (vì Cassandra có hầu hết các đặc tính của NoSQL)***
* Dữ liệu trong NoSQL DB được lưu dưới dạng document, object. Truy vấn dễ dàng và nhanh hơn RDBMS nhiều.
* NoSQL có thể hoàn toàn làm việc tốt với dữ liệu dạng không có cấu trúc.
* Việc đổi cấu trúc dữ liệu (thêm, xóa trường hoặc bảng) rất dễ dàng và nhanh gọn trong NoSQL.
* Vì không đặt nặng tính ACID của transactions và tính nhất quán của dữ liệu, NoSQL DB có thể mở rộng, chạy trên nhiều máy một cách dễ dàng.

1. **Vì sao nên sử dụng Cassandra**

Chúng ta nên sử dụng Cassandra vì những đặc tính nổi bật sau:

* Chấp nhận sai sót (Fault Tolerant): Dữ liệu của bạn đẽ được sao chép thành nhiều bản trên các server. Nếu chẳng may một server nào đó bị hỏng, bạn vẫn có thể truy xuất dữ liệu của bạn trên các server khác.
* Tính co giãn (Elastic): Khả năng đọc/ghi tăng tuyến tính theo số lượng máy được thêm vào cụm máy của bạn mà không có thời gian chết (downtime) hay sự gián đoạn ứng dụng đang chạy của bạn.
* Hướng cột (Column-Oriented): Các RDBMS hướng dòng (row-oriented) phải định nghĩa trước các cột (column) trong các bảng (table). Đối với Cassandra các bạn không phải làm điều đó, đơn giản là thêm vào bao nhiêu cột cũng được tùy theo nhu cầu của bạn.
* Tính sẵn sàng cao (Highly Availability): Khi thực hiện tác vụ đọc/ghi, Cassandra có thể thực hiện trên bản sao gần nhất hoặc trên tất cả các bản sao. Điều này phụ thuộc vào thông số ConsitencyLevel do bạn thiết lập.
* Kiến trúc ngang hàng: Tất cả các node trong cassandra cluster có vai trò tương tự nhau, đều đảm nhận việc đọc và ghi dữ liệu, Cassandra được đánh giá nổi bật là dựa trên yếu tố này, xây dựng kiến trúc ngang hàng sẽ làm giảm nguy cơ bị thắt cổ chai (bottlenect)

1. **Những trường hợp nên và không nên sử dụng Cassandra**

* Bởi vì Cassandra có khả năng xử lý tốt một khối lượng lớn dữ liệu nên nó được sử dụng trong rất nhiều các ứng dụng khác nhau, từ những ứng dụng phù hợp với xử lý về mặt tốc độ, đến việc kết hợp với các công nghệ khác trong việc xử lý thời gian thực của BigData.
* Cassandra không nên được dùng cho những dữ liều cần xử lý, tính toán nhiều trên storage, nó không hỗ trợ các hàm Sum, Group, Join, Max, Min và bất kỳ hàm nào khác developer muốn sử dụng để tính toán dữ liệu khi query. Cassandra cũng không hỗ trợ build-in full text search, B-tree index hoặc các hàm trợ giúp tính toán.