Chương 13

CƠ SỞ DỮ LIỆU NOSQL

Nội dung

- Khái niệm NoSQL
- Đặc điểm
- Phân loại và ví dụ
- Ưu và nhược điểm
- Một số hệ quản trị NoSQL phổ biến

Khái niệm NoSQL

- Cơ sở dữ liệu **NoSQL** (tên gốc là "Non SQL" (phi SQL) hoặc "non relational" (phi quan hệ))[1] cung cấp một cơ chế để lưu trữ và truy xuất dữ liệu được mô hình hóa khác với các quan hệ bảng được sử dụng trong các cơ sở dữ liệu kiểu quan hệ.
- Một mệnh đề khá thú vị về NoSQL:
  - "select fun, profit from real_world where relational=false;"

Lịch sử

- *NoSQL* được sử dụng bởi Carlo Strozzi vào năm 1998 để đặt tên cho cơ sở dữ liệu quan hệ mã nguồn mở Strozzi NoSQL nhỏ gọn của mình, mà không tiết lộ giao diện SQL tiêu chuẩn, nhưng vẫn còn là kiểu quan hệ. Strozzi gọi là 'NoREL'",nghĩa là "No Relational'.
- Đầu năm 2009 khi tổ chức một sự kiện thảo luận về "các cơ sở dữ liệu phân tán, không quan hệ nguồn mở". Johan Oskarsson của Last.fm giới thiệu lại thuật ngữ NoSQL là distributed (phân tán) + non-relational (không ràng buộc).

Lịch sử

- NoSQL đánh dấu sự xuất hiện ngày càng nhiều các kho lưu trữ dữ liệu phân tán, không quan hệ, bao gồm các nhân bản mã nguồn mở BigTable/MapReduce của Google và Dynamo của Amazon. Hầu hết các hệ thống NoSQL đầu tiên đã không cố gắng cung cấp các bảo đảm tính nguyên tố, nhất quán, tách biệt và bền vững, trái với ưu thế thực tế trong các hệ thống cơ sở dữ liệu kiểu quan hệ.
- Dựa trên doanh thu năm 2014, các hãng dẫn đầu thị trường NoSQL là MarkLogic, MongoDB, và Datastax.
- Dựa trên các bảng xếp hạng phổ biến năm 2015, Các cơ sở dữ liệu NoSQL phổ biến nhất là MongoDB, Apache Cassandra, và Redis.

Đặc điểm NoSQL

- NoSQL lưu trữ dữ liệu của mình theo cặp giá trị "key-value". Sử dụng lớn các node để lưu trữ thông tin.
- Chấp nhận dữ liệu bị trùng lặp do một số node lưu cùng thông tin giống nhau.
- Phi quan hệ: không có ràng buộc nào cho việc nhất quán dữ liệu.
- Tính nhất quán không theo thời gian thực.

Đặc điểm NoSQL

Xử lý dữ liệu quan hệ

- Do hầu hết các cơ sở dữ liệu NoSQL thiếu khả năng kết nối trong các truy vấn, lược đồ cơ sở dữ liệu nói chung cần phải được thiết kế khác nhau.Có ba kỹ thuật chính để xử lý dữ liệu quan hệ trong một cơ sở dữ liệu NoSQL.

Đặc điểm NoSQL

Đa truy vấn:

- Thay vì lấy tất cả các dữ liệu với một truy vấn, ta thường thực hiện nhiều truy vấn khác nhau để có được các dữ liệu mong muốn. Các truy vấn NoSQL thường nhanh hơn so với truy vấn SQL truyền thống vì vậy chi phí của việc phải thực hiện các truy vấn bổ sung có thể chấp nhận được. Nếu số lượng truy vấn quá nhiều là cần thiết, một trong hai phương pháp khác sẽ thích hợp hơn.

Đặc điểm NoSQL

Dữ liệu bộ nhớ đệm/sao chép/không-chuẩn hoá

- Thay vì chỉ lưu giữ các từ khóa ngoại lai, ta thường lưu trữ các giá trị thực tế ngoại lai cùng với dữ liệu của mô hình.
- Ví dụ, mỗi bình luận blog có thể bao gồm tên người dùng, thêm vào đó là một id người dùng, do đó ta dễ dàng truy cập đến tên người dùng mà không cần phải có bất kỳ tra cứu nào khác. Khi một tên người dùng thay đổi tuy nhiên, điều này giờ đây sẽ cần phải được thay đổi ở nhiều nơi trong cơ sở dữ liệu. Vì vậy phương pháp này hoạt động tốt hơn khi việc đọc là phổ biến hơn nhiều so với việc ghi.

Đặc điểm NoSQL

Nesting data

- With document databases like MongoDB it's common to put more data in a smaller number of collections. For example, in a blogging application, one might choose to store comments within the blog post document so that with a single retrieval one gets all the comments. Thus in this approach a single document contains all the data you need for a specific task

Đặc điểm NoSQL

Hỗ trợ ACID và JOIN

Cơ sở dữ liệu	ACID	Joins
Aerospike	Có	Không
ArangoDB	Có	Có
CouchDB	Có	Có
c-treeACE	Có	Có
HyperDex	Có[nb 1]	Có
InfinityDB	Có	Không
LMDB	Có	Không
MarkLogic	Có	Có[nb 2]
OrientDB	Có	Có

Phân loại – Các dạng NoSQL cơ bản

- Key – value data stores: dữ liệu lưu dưới dạng cặp key-value. Giá trị được truy xuất thông qua key.
- Column – based – Tabular (Column Families Stores): cơ sở dữ liệu tổ chức dưới dạng bảng.
- Document – based: dữ liệu được lưu trữ và tổ chức dưới dạng một tập hợp các document.
- Graph – based data – stores: áp dụng lý thuyết đồ thị trong khoa học máy tính để lưu trữ và truy xuất dữ liệu, tập trung vào tính rời rạc giữa các phần dữ liệu.

Key – value data stores

- Kho lưu trữ khóa-giá trị (Key-value: KV) sử dụng mảng kết hợp (còn được gọi là bản đồ hoặc từ điển).
- Một bảng băm hay một danh sách liên kết là cấu trúc dữ liệu đơn giản nhất có thể chứa tập hợp các cặp khoá – giá trị (key-values).
- Độ phức tạp trung bình O(1) cho các thuật toán truy cập dữ liệu.
- Khoá của 1 cặp key-values là giá trị duy nhất trong các bộ và dễ dàng tìm kiếm để truy cập dữ liệu.

Key – value data stores

- Cặp key-values có nhiều kiểu:
  - Lưu trữ dữ liệu trong bộ nhớ RAM.
  - Lưu trữ dữ liệu trực tiếp trong ổ cứng.
  - Lưu trữ dữ liệu trong bộ nhớ cache. (phổ biến hơn cả)
- Một số cơ sở dữ liệu NoSQL tiêu biểu cho dạng này: Oracle, DynamoDB, Redis, FoundationDB, BerkeleyDB, Voldermort,…

Column Families Stores

- Mô hình mà dữ liệu được lưu trữ định hướng cột được khởi xướng bởi Google (BigTable).
- Mỗi đơn vị dữ liệu có thể được coi như một tập hợp các cặp khoá – giá trị, trong đó mỗi đơn vị được xác định bằng một định danh chính, giống như là khoá chính, và có xu hướng được gọi là row-key.

Column Families Stores

Ví dụ về row-key

For row-key: 1	For row-key: 2
first_name: John	first_name: Jane
last_name: Doe	
zip_code: 10001	zip_code: 94303
gender: male	

- Chỉ có cặp khoá – giá trị hợp lệ được lưu trữ
  - Một column-family ‘name’ với hàng first_name và last_name.
  - Một column-family khác ‘location’ với zip_code.
  - Column-family ‘profile’ với hàng gender.
- Column-family được định nghĩa khi cấu hình hoặc lúc bắt đầu chạy.
- Các cột có khả năng lưu trữ bất kỳ loại dữ liệu nào.

Column Families Stores

- Dữ liệu được lưu trữ theo trình tự liên tiếp.
- Dữ liệu tồn tại được chịu lỗi bằng cách tạo ra 3 bản sao của mỗi bộ dữ liệu được duy trì.
- Hệ thống cho phép dữ liệu được lưu trữ trong 1 cụm máy (lưu trữ phân tán).
- Một số cơ sở dữ liệu NoSQL tiêu biểu cho dạng này: Hadoop/Hbase, Cassandra, Hypertable, Cloudata,…

Document Stores

- Đây không phải là một hệ thống quản lý văn bản.
- Các document trong Document Stores bao gồm cấu trúc key-values khá lỏng lẻo.
  - Định dạng JSON (JavaScript Object Notatin).
  - Không có tài liệu hay bảng tính (mặc dù chúng có thể được lưu trữ).

Document Stores

- Document Stores xử lý toàn bộ một document một lúc và tránh việc cắt nhỏ bản ghi thành các giá trị key-values cấu thành nên nó.
- Ở mức độ cao hơn, nó cho phép đặt các bản ghi có chung đặc điểm vào một bộ dữ liệu duy nhất (gọi là Collections).
- Document Stores cho phép lập chỉ mục cho các bản ghi trên cơ sở không chỉ đích danh khoá chính mà là tất cả các chỉ mục bên trong của nó.

Graph Databases/Stores

- Đa số cơ sở dữ liệu NoSQL thuộc 3 dạng trên.
- Số ít còn lại là cơ sở liệu dạng biểu đồ (điểm, cạnh, đường) và cơ sở dữ liệu lưu trữ dưới dạng XML. Chúng cũng được coi là cơ sở dữ liệu NoSQL.
- Một số ít cơ sở dữ liệu NoSQL thuộc dạng này tiêu biểu như: Neo4j và FlockDB.

Graph Databases/Stores

- Loại cơ sở dữ liệu này được thiết kế cho dữ liệu có quan hệ cũng được biểu diễn như một đồ thị bao gồm các yếu tố kết nối qua lại với một số hữu hạn các quan hệ giữa chúng.
- Loại dữ liệu này có thể là các mối quan hệ xã hội, liên kết giao thông công cộng, bản đồ đường bộ hoặc các topo mạng

Graph Databases/Stores

- Loại cơ sở dữ liệu này được thiết kế cho dữ liệu có quan hệ cũng được biểu diễn như một đồ thị bao gồm các yếu tố kết nối qua lại với một số hữu hạn các quan hệ giữa chúng.
- Loại dữ liệu này có thể là các mối quan hệ xã hội, liên kết giao thông công cộng, bản đồ đường bộ hoặc các topo mạng

Graph Databases/Stores

Tên	Ngôn ngữ	Ghi chú
AllegroGraph	SPARQL	RDF triple store
DEX/Sparksee	C++, Java, .NET, Python	Graph database
FlockDB	Scala	Graph database
IBM DB2	SPARQL	RDF triple store added in DB2 10
InfiniteGraph	Java	Graph database
MarkLogic	Java, JavaScript, SPARQL, XQuery	Multi-model document database and RDF triple store
Neo4j	Cypher	Graph database
OpenLink Virtuoso	C++, C#, Java, SPARQL	Middleware and database engine hybrid
Oracle	SPARQL 1.1	RDF triple store added in 11g
OrientDB	Java	Multi-model document and graph database
OWLIM	Java, SPARQL 1.1	RDF triple store
Sqrrl Enterprise	Java	Graph database

KIẾN TRÚC NOSQL

- Sơ lược
- Kiến trúc lưu trữ
- Các đặc điểm chung
- NoSQL với các ngôn ngữ lập trình bậc cao
- Ưu, nhược điểm của NoSQL

Sơ lược

- Thiết kế trên nguyên tắc Distributed NoSQL giảm thiểu tối đa các phép tính toán không cần thiết.
- Tối thiểu hoá thao tác đọc-ghi dữ liệu trực tiếp kết hợp với batch processing.
- Mô hình lưu trữ tập dữ liệu theo cặp giá trị key-values.
- Quản lý dữ liệu phân tán dưới dạng note (nút), chấp nhận việc trùng lặp dữ liệu.

Kiến trúc lưu trữ NoSQL

- NoSQL Column Families
- NoSQL Document Stores

NoSQL Column Families (1)

- Trong một NoSQL Column Families, các column-family giống như các cột ở hệ thống RDBMS.
- Chúng cùng được định nghĩa trước khi dữ liệu được lưu trữ và thường ít thay đổi trong quá trình cập nhật.
- Điểm khác biệt giữa chúng là khi khai báo cột bên hệ thống RDBMS thì phải khai báo loại dữ liệu có thể lưu trữ (INT, VARCHAR, TEXT,…), còn NoSOL Column Families

NoSQL Column Families (2)

- Thường áp đặt một lược đồ (schema) đơn giản trước và có thể dễ dàng tạo thêm các cột mới.
- Một column-family là một tập hợp của các cột nhỏ hơn, thường thì các cột nhỏ này có quan hệ logic với nhau. Chúng cũng được lưu trữ vật lý cùng nhau.

NoSQL Column Families (3)

- Trong lưu trữ vật lý, dữ liệu không được lưu trữ theo một "bảng" duy nhất mà được lưu thành nhiều column-family trong bảng.
- Do cấu trúc phân tán nên một bảng có thể nằm ở nhiều máy chủ khác nhau, nhưng cùng được xác định bởi một row-key duy nhất.

NoSQL Column Families (4)

Ví dụ về lưu trữ column-families

Table 'people'

```
{
        "row_key_1" : {
                "name" : {
                ...
                },
                "location" : {
                ...
                },
                "preferences" : {
                ...
                }
        },
        "row_key_2" : {
                "name" : {
                ...
                },
                "location" : {
                ...
                },
                "preferences" : {
                ...
                }
        },
        "row_key_3" : {
                "name" : {
                ...
                },
                "location" : {
                ...
                },
                "preferences" : {
                ...
                }
        }
}
```

```
{
        "row_key_1" : {
                "name" : {
                ...
        },
        "row_key_2" : {
                "name" : {
                ...
                }
        "row_key_3" : {
                "name" : {
                ...
        }
}
```

NoSQL Document Stores (1)

- Collections trong Document Stores được coi như bảng bên các hệ thống RDBMS, tuy nhiên các Collections không có các ràng buộc về mặt quan hệ như các bảng RDBMS.
- Trong các Collections này lưu trữ các document.
- Mỗi document được lưu dưới dạng dữ liệu JSON (ở MongoDB là dạng BSON – Binary JSON) và thường thì chúng hỗ trợ mọi kiểu dữ liệu thường gặp.
- Các NoSQL Document Stores hầu hết tuân theo định dạng trên để lưu trữ dữ liệu nhưng với mỗi cơ sở dữ liệu lại có một cách lưu trữ vật lý khác nhau.

NoSQL Document Stores (2)

- MongoDB là một ví dụ điển hình cho NoSQL Document Stores và tuân thủ chính xác những định dạng trên.
- MongoDB lưu trữ vật lý dạng memory-mapped (sử dụng bộ nhớ cache).
- Một tập tin memory-mapped là một phân đoạn của bộ nhớ ảo, có vai trò tương tác gián tiếp (tham chiếu) với tập tin nằm trên ổ cứng.
- Điều này thực sự cải thiện tốc độ đọc-ghi so với đọc ghi thông thường.

Đặc điểm chung

- High Scalability: Độ mở rộng cao
- High Availability: Tỉ lệ "chết" của hệ thống rất thấp
- Atomicity: Độc lập data state trong các operation.
- Consistency: tính nhất quán yếu.
- Deployment Flexibility: hệ thống tự động sửa lỗi lưu trữ mà không cần can thiệp.
- Query Flexibility: Multi-Gets, Range queries

NoSQL với các ngôn ngữ lập trình

- Sử dụng giao diện dòng lệnh là một cách nhanh và hiệu quả với các truy vấn cơ sở dữ liệu NoSQL, thế nhưng trong một dự án thực tế, không phải ai cũng có thể sử dụng nhưng giao diện này một cách thông thạo và nhuần nhuyễn mà cần tới một giao diện trực quan. Chính vì điều này, nên có rất nhiều các thư viện hỗ trợ thực thi NoSQL với các ngôn ngữ lập trình bậc cao như Java, Python, Ruby hay PHP,… Những thư viện này cung cấp những

NoSQL với các ngôn ngữ lập trình

- Framework Thrift
- NoSQL với Java
- NoSQL với Python
- NoSQL với PHP

Framework Thrift

- Thrift là một framework phát triển ứng dụng mã nguồn mở. Nó giống như một ứng dụng nền để xây dựng các dịch vụ giao tiếp từ NoSQL với nhiều ngôn ngữ lập trình khác nhau. Với mỗi ngôn ngữ lập trình khác nhau, cần phải cấu hình Thrift theo một cách khác nhau.
- Thrift được xây dựng bởi Facebook và sau đó được mở rộng bởi cộng đồng mã nguồn mở trên thế giới. Thrift được viết bằng ngôn ngữ C.
- Việc sử dụng Thrift đem đến khá nhiều hiệu quả do khả năng tương thích với nhiều ngôn ngữ bậc cao. Nhưng không nhất thiết phải sử dụng Thrift vì các cơ sở dữ liệu NoSQL có hỗ trợ các ngôn ngữ lập trình cụ thể.

NoSQL với Java (1)

- Java là một ngôn ngữ lập trình rất phổ biến và có độ ứng dụng rộng rãi. Hầu hết các cơ sở dữ liệu NoSQL hỗ trợ Java, trong đó có MongoDB và Hbase.
- MongoDB chính thức hỗ trợ Java và phân phối một thư viện riêng dành cho việc kết nối với Java.

NoSQL với Java (2)

- Thư viện và tài liệu về thư viện này có sẵn trên địa chỉ http://mongodb.org/display/DOCS/Java+Language+Center.
- Thư viện này được đóng gói dưới dạng .jar. Sau khi tải về, chỉ cần thêm đường dẫn tới thư viện này vào trong classpath của ứng dụng là có thể sử dụng.
- Sau đây là một ví dụ đơn giản về việc truy xuất tới cơ sở dữ liệu MongoDB với Java.

NoSQL với Java (3)

- Chương trình bên dung Java kết nối đến cơ sở dữ liệu mydb và lấy toàn bộ dữ liệu trong bảng ‘logdata’ và in ra ngoài màn hình

javaMongoDBClient.java

```
import com.mongodb.DB;
import com.mongodb.DBCollection;
import com.mongodb.DBCursor;
import com.mongodb.Mongo;

public class JavaMongoDBClient {
        Mongo m;
        DB db;
        DBCollection coll;
        public void init() throws Exception {
                m = new Mongo( “localhost” , 27017 );
                db = m.getDB( “mydb” );
                coll = db.getCollection(“logdata”);
        }
        public void getLogData() {
                DBCursor cur = coll.find();
                while(cur.hasNext()) {
                        System.out.println(cur.next());
                }
        }
        public static void main(String[] args) {
                try {
                        JavaMongoDBClient javaMongoDBClient;
                        javaMongoDBClient  = new JavaMongoDBClient();
                        javaMongoDBClient.init();
                        javaMongoDBClient.getLogData();
                        } catch(Exception e) {
                                e.printStackTrace();
                        }
                }
}
```

NoSQL với Python (1)

- Python là một ngôn ngữ lập trình mạnh và khá mềm dẻo. Ta hoàn toàn có thể sử dụng Python trong việc truy vấn đến các cơ sở dữ liệu NoSQL bằng các thư viện mà chính các cơ sở dữ liệu này hỗ trợ.
- Ví dụ như với NoSQL Apache Cassandra, có thể sử dụng thư viện Pycassa để thực hiện truy vấn. Thư viện này được cung cấp trực tiếp tại trang http://github.com/pycassa/pycassa.

NoSQL với Python (2)

- Sau khi tải về và cài đặt vào trong thư viện của Python, ta có thể dễ dàng sử dụng thư viện này bằng cách:
  - *import pycassa*
- Một số câu lệnh truy vấn khác, như:
  - Tạo kết nối đến cơ sở dữ liệu: *connection = pycassa.connect('database')*
  - Sử dụng column-families 'people':
    - *column_family = pycassa.ColumnFamily(connection, 'post')*

NoSQL với PHP (1)

- PHP là một ngôn ngữ được sử dụng rất nhiều trong thiết kế và phát triển các ứng dụng Web. Ngoài việc tương thích rất tốt với các cơ sở dữ liệu quan hệ RDBMS, PHP cũng được rất nhiều các cơ sở dữ liệu NoSQL hỗ trợ.
- Giống như Python, Apache Cassandra cũng hỗ trợ PHP với một thư viện có tên phpcassa. Thư viện này được phân phối dạng mã nguồn mở tại địa chỉ http://github.com/hoan/phpcassa.

NoSQL với PHP (2)

phpcassa_examle.php

```
<?php
 // Copy all the files in this repository to your include directory.
$GLOBALS['THRIFT_ROOT'] = dirname(__FILE__) . '/include/thrift/';
require_once $GLOBALS['THRIFT_ROOT'].'/packages/cassandra/Cassandra.php';
require_once $GLOBALS['THRIFT_ROOT'].'/transport/TSocket.php';
require_once $GLOBALS['THRIFT_ROOT'].'/protocol/TBinaryProtocol.php';
require_once $GLOBALS['THRIFT_ROOT'].'/transport/TFramedTransport.php';
require_once $GLOBALS['THRIFT_ROOT'].'/transport/TBufferedTransport.php';
include_once(dirname(__FILE__) . '/include/phpcassa.php');
include_once(dirname(__FILE__) . '/include/uuid.php');
$posts = new CassandraCF('database', 'post'); // Create a connect with 'database'
$posts ->get(); // Get all data
?>
```

Ưu điểm của NoSQL

- Hiệu suất hoạt động cao. Dễ dàng mở rộng quy mô dữ liệu.
- Giải quyết tốt các cơ sở dữ liệu có kích thước và truy cập lớn.
- Không thực sự cần các DBAs (Database Administrators).
- Có hiệu quả kinh tế cao.
- Mô hình dữ liệu linh hoạt.
- NoSQL phù hợp với công nghệ đám mây

Nhược điểm của NoSQL

- Khả năng tin cậy do cấu trúc dữ liệu phi quan hệ
- Sự hỗ trợ bên phía nhà sản xuất. Nguồn mở có nghĩa là sự hỗ trợ không đồng đều cho các doanh nghiệp
- Những vấn đề về tính tương thích
- Quản trị.
- Chuyên môn của người quản lý.
- Thiếu sự tinh thông

Một số hệ quản trị NoSQL phổ biến

- MongoDB và Redis: lưu trữ các dữ liệu thống kê ít được đọc, được viết thường xuyên.
- Hadoop: một CSDL dạng tự do, lưu trữ các dữ liệu lớn như các con số thống kê thời tiết hoặc công việc phân tích nghiệp vụ.
- Memcache: lưu trữ các phiên làm việc web, các khóa, và các con số thống kê ngắn hạn.
- Cassandra và Riak: làm việc tốt trong các môi trường với các ứng dụng có tính sẵn sàng cao, khi thời gian

APACHE CASSANDRA

- Đặc điểm của Cassandra
- Kiến trúc của Cassandra
- Mô hình dữ liệu
- Cài đặt Cassandra
- Demo Cassandra và so sánh với MySQL

Đặc điểm của Cassandra

- Tính phân tán và không tập trung hoá.
- Khả năng mở rộng mềm dẻo.
- Tính sẵn sàng cao và khả năng chịu lỗi.
- Tính nhất quán cuối.
- Hướng cột
- Schema – Free
- Hiệu năng cao

Kiến trúc của Cassandra

- Kết nối giữa các nút
- Các thành viên của cụm và các nút hạt giống
- Trạng thái dò thất bại và phục hồi
- Phân vùng dữ liệu trong Cassandra
- Nhân bản trong Cassandra
- Snitches

Kết nối giữa các nút

- Gossip là một giao thức truyền thông ngang hàng, trong đó các nút định kỳ trao đổi thông tin trạng thái về bản thân và về các nút khác mà chúng biết.
- Trong Cassandra, quá trình Gossip chạy mỗi giây và mỗi nút trao đổi các thông báo trạng thái của nó tối đa với 3 nút khác trong cluster. Các nút thông tin trao đổi về bản thân và về các nút khác mà họ nó đã được biết, do đó, tất cả các nút nhanh chóng tìm hiểu về tất cả các

Các thành viên của cụm và các nút hạt giống

- Khi một nút bắt đầu, nó nhìn vào tập tin cấu hình của nó để xác định tên của cụm Cassandra chứa nó và nút được gọi là hạt giống để liên hệ, có được thông tin về các nút khác trong cluster. Điểm liên lạc của các cụm được cấu hình trong file cassandra.yaml cho mỗi nút.
- Tất cả các nút trong cluster phải có cùng một danh sách các nút hạt giống được liệt kê trong tập tin cấu hình của nó.

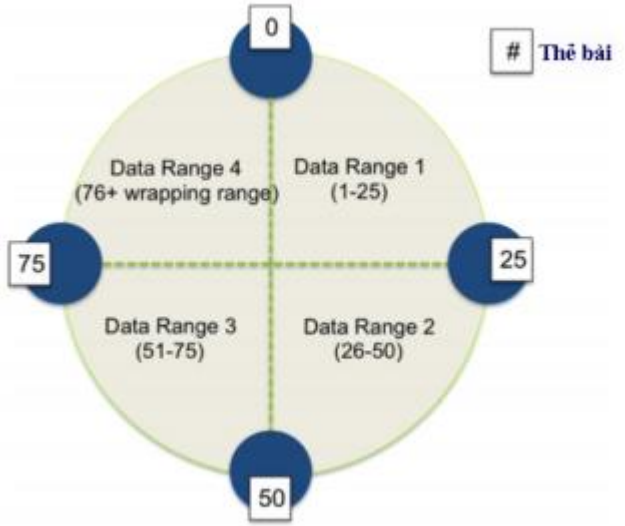
Trạng thái dò thất bại và phục hồi

- Dò thất bại là thông qua trạng thái của tin đồn (gossip), từ 1 nút xác định xem các nút khác trong hệ thống đang online hay offline. Thông tin dò thất bại cũng được sử dụng trong Cassandra để tránh định tuyến các yêu cầu từ máy khách đến các nút không thể truy cập được.

Phân vùng dữ liệu trong Cassandra (1)

- Khi khởi động một cụm Cassandra, người dùng phải chọn cách thức dữ liệu sẽ được chia trên các nút trong cluster.
- Trong Cassandra, tổng số dữ liệu được quản lý bởi 1 cụm được đại diện như là một không gian hoặc vòng tròn.

Phân vùng dữ liệu trong Cassandra (2)



Nhân bản trong Cassandra

- Là quá trình lưu trữ các bản sao của dữ liệu trên nhiều nút để đảm bảo độ tin cậy và khả năng chịu lỗi.
- Tổng số bản sao trên cluster thường được gọi là nhân tố nhân bản.
- Chiến lược xác định vị trí nhân bản :
  - SimpleStrategy
  - NetworkTopologyStrategy

Snitches

- Snitch là một thành phần cấu hình của một cụm Cassandra được sử dụng để xác định các nút được nhóm lại với nhau như thế nào trong cấu trúc liên kết mạng tổng thể (rack và các nhóm trung tâm dữ liệu).
- Snitches được cấu hình cho một cụm Cassandra trong file cấu hình cassandra.yaml.

Mô hình dữ liệu Cassandra

- Keyspaces
- Column-Families
- Columns

Keyspaces

- Trong Cassandra, keyspace là nơi chưa dữ liệu cho ứng dụng, giống như lược đồ trong một cơ sở dữ liệu quan hệ.
- Dùng để nhóm các column family lại với nhau.
- Việc nhân bản được điều khiển trên cơ sở keyspaces.
- Keyspace không được thiết kế để sử dụng như một lớp bản đồ quan trọng trong mô hình dữ liệu, mà nó chỉ như một cách để điều khiển việc nhân bản dữ liệu cho một tập các column family.

Column-Families

- Là một tập hợp các cột tương đương
- Mỗi colimn-families có số lượng cột có thể khác nhau.
- Có 2 loại Column-Families:
  - Column-Families tĩnh
  - Column-Families động

Columns

- Cột là đơn vị dữ liệu nhỏ nhất trong Cassandra. Nó là một bộ gồm có tên, giá trị, và một nhãn thời gian.
- Cột có thể được đánh chỉ mục theo tên của nó, không nhất thiết phải có một giá trị.
- Cassandra sử dụng cột nhãn thời gian để xác định cập nhật gần nhất của một cột.

Cài đặt Cassandra

- Cài đặt trên hệ điều hành Ubuntu 13.04 :
  - Yêu cầu hệ thống: Cài đặt Java 7, tốt nhất là Oracle/Sun JVM. Cassandra cũng chạy tốt với OpenJDK và IBM JVM. Cassandra không chạy trên JRockit.
  - Cách cài đặt:
    - Download phiên bản Cassandra tại trang chủ http://cassandra.apache.org/download/
    - Giải nén
    - Cài đặt các thư mục data_file_directory, commit_log_directory, saved_caches_directory
    - Khởi động Cassandra bằng câu lệnh 'bin/cassandra -f' trên command line.

Làm việc với Cassandra (1)

- Sử dụng công cụ Cassandra-cli :
  - Nằm trong thư mục Cassandra/bin .
    - cassandra-cli –host localhost –port 9160
    - cassandra-cli –host 110.123.4.5 –port 9160

Làm việc với Cassandra (2)

- Ví dụ các câu lệnh
  - Tạo keyspace
    - CREATE KEYSPACE demo WITH PLACEMENT_STRATEGY='org.apache.Cassandra.locator.SimpleStrategy'

    and strategy_options = [{replication_factor:1}];

Làm việc với Cassandra (3)

- Ví dụ các câu lệnh
  - Tạo column family
    - CREATE COLUMN FAMILY users
    - WITH comparator = UTF8Type
    - AND key_validation_class=UTF8Type
    - AND column_metadata = [
    - {column_name: full_name, validation_class: UTF8Type}
    - {column_name: email, validation_class: UTF8Type}
    - {column_name: state, validation_class: UTF8Type}
    - {column_name: gender, validation_class: UTF8Type}
    - {column_name: birth_year, validation_class: LongType}
    -];

Làm việc với Cassandra (4)

- Ví dụ các câu lệnh
  - Thêm hàng và cột
  - SET users['bobbyjo']['full_name']='Robert Jones';
  - SET users['bobbyjo']['email']='bobjones@gmail.com';
  - SET users['bobbyjo']['state']='TX';
- Đọc dữ liệu
  - LIST users;
  - GET users[utf8('bobbyjo')][utf8('full_name')];

Thực hiện một số truy vấn trên Cassandra

DEMO CASSANDRA

CẢM ƠN!