BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÁCH KHOA TP HCM
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

- - - - - - * - - - - - -

Data Engineering

Báo cáo kỹ thuật cho bài kiểm tra cuối kỳ

GV Hướng dẫn: PGS. TS. ĐẶNG TRẦN KHÁNH
HV Thực hiện: 2070682 – PHẠM NGUYỄN NHẬT MINH

Tp. Hồ Chí Minh, Tháng 12/2021

Giới thiệu mục tiêu báo cáo

Bigtable là một trong những cơ sở dữ liệu dạng đồ thị phổ biến hiện nay, được giới thiệu lần đầu vào năm 2004 và đến nay đã được sử dụng trong rất nhiều ứng dụng của Google. Mục tiêu của báo cáo này là giới thiệu cái nhìn tổng quan về cơ chế hoạt động của Bigtable và xây dựng một ứng dụng minh họa cho khả năng của công nghệ này với dữ liệu thực tiễn.

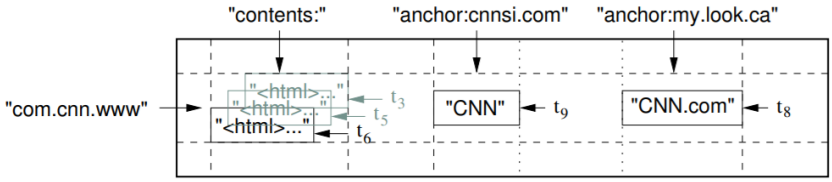
Cơ sở lý thuyết

Những thay đổi về công nghệ cũng như định dạng dữ liệu trong những năm gần đây đã có nhiều tác động đến những công nghệ về cơ sở dữ liệu. Những ứng dụng hiện đại ngày càng sử dụng những dữ liệu có tính kết nối cao như những kiến trúc đồ thị, văn bản, siêu văn bản, RDF hay mạng xã hội. Những cơ sở dữ liệu truyền thống (RDBMS) đã được nghiên cứu và chứng minh là không hiệu quả [1] trong các vấn đề về mở rộng (scalability) hay tính linh hoạt trong xử lý những loại dữ liệu trên. Cơ sở dữ liệu phi cấu trúc hay cơ sở dữ liệu dạng đồ thị được sinh ra để giải quyết bài toán này.

Cơ sở dữ liệu đồ thị [2] được xây dựng trên nền tảng lý thuyết đồ thị, với các thuật ngữ và đối tượng trừu tượng: đỉnh (nodes), cạnh (edges) và các thuộc tính của chúng. Cho phép người dùng định nghĩa các tác vụ thao tác tùy theo mục đích sử dụng trên dữ liệu được tối ưu thông qua việc sử dụng các giải thuật đồ thị.

Tương ứng với những sự phát triển về dữ liệu, số lượng các cơ sở dữ liệu đồ thị được công bố ngày càng gia tăng. Trong số đó, Bigtable là một hệ cơ sở dữ liệu được phát triển bởi Google với những đặc tính nổi trội bao gồm: tốc độ xử lý nhanh, lưu trữ dữ liệu theo cấu trúc phân tán phi tập trung và khả năng mở rộng với dữ liệu lên đến hàng pentabytes trên hàng ngàn node máy trạm với kiến trúc cloud của Google [5]. Bigtable đã chứng minh được khả năng cung cấp một giải pháp linh hoạt, hiệu năng cao bằng việc được sử dụng trong hơn 60 sản phẩm khác nhau của Google. Có thể kể đến một vài dự án đòi hỏi yêu cầu về kích thước dữ liệu rất lớn như web indexing của dịch vụ Google search, Google Analytics hay đòi hỏi độ trễ thấp như Google Maps, Google Earth [6].

Bigtable được xây dựng với triết lý là một cơ sở dữ liệu bán cấu trúc, lưu trữ dữ liệu dạng key-value. Có thể xem Bigtable như một cây chỉ mục lớn, mà một index key được xây dựng từ bộ ba column key: *String*, row key: *String* và timestamp: *Int64*. Value của cây chỉ mục chính là dữ liệu thực được lưu trữ, mỗi tác vụ đọc ghi trên dữ liệu là atomic. Một đặc tính khác của Bigtable đó là dữ liệu có tính thưa thớt (sparse), mỗi hàng dữ liệu sẽ sử dụng những cột dữ liệu khác nhau và phần lớn các cột dữ liệu trong một hàng sẽ là rỗng. Toàn bộ dữ liệu sẽ được chia nhỏ theo hàng cho mục đích phân tán và hàng là đơn vị dữ liệu phân chia nhỏ nhất. Một máy chủ sẽ quản lý vị trí của các hàng dữ liệu trong hệ thống cũng như các hàng liền kề.



Hình 1: *Một hàng trong dữ liệu web indexing, tên cột là nghịch đảo của địa chỉ trang web, cột content cho biết nội dung trang web qua các version (tương ứng với các timestamp khác nhau), cột anchor cho biết trang web tham khảo đến trang web CNN [6]*

Hiện thực

Báo cáo này đã hiện thực lại một hệ thống giúp truy xuất dữ liệu giao thông sử dụng công nghệ Bigtable để minh họa cho khả năng của Bigtable đối với dữ liệu thực tiễn. Dữ liệu được sử dụng là một dữ liệu nổi tiếng của dịch vụ xe buýt thành phố New York với 300 tuyến đường và 5600 phương tiện trong vòng 1 tháng [7].

Để đạt được hiệu suất cao nhất cho Bigtable thì việc thiết kế lược đồ cho cơ sở dữ liệu là quan trọng nhất. Các dữ liệu được lưu trong cơ sở dữ liệu đều được sắp xếp một cách tự động, do đó việc thiết kế thích hợp sẽ giúp truy xuất những cụm dữ liệu liên quan được tối ưu. Ngoài ra cũng cần phải tránh tình trang tập trung các cụm dữ liệu lại quá gần nhau, như việc thiết kế khóa hàng bắt đầu bằng timestamp. Việc này làm sinh ra những điểm nóng dữ liệu, khiến cho việc cân bằng tải trở nên khó khăn hơn và không tận dụng tối đa được khả năng tính toán song song.

Row key	cf:VehicleLocation.Latitude	cf:VehicleLocation.Longitude	...
MTA/M86-SBS/1496275200000/NYCT_5824	40.781212 *@20:52:54.00*40.776163 *@20:43:19.00*40.778714 *@20:33:46.00*	-73.961942 *@20:52:54.00*-73.946949 *@20:43:19.00*-73.953731 *@20:33:46.00*	...
MTA/M86-SBS/1496275200000/NYCT_5840	40.780664 *@20:13:51.00*40.788416 *@20:03:40.00*	-73.958357 *@20:13:51.00* -73.976748 *@20:03:40.00*	...

Hình 2: *Lược đồ sử dụng [Bus company/Bus line/Timestamp in hour/Vehicle ID] làm row key*

Kết quả cụ thể sẽ được trình bày trong phần phụ lục của báo cáo. Ngoài ra, mã nguồn và dữ liệu được sử dụng trong báo cáo có thể được tìm thấy và tải xuống ở đường dẫn sau: https://github.com/hacba98/de-final-project

Kết luận

Trong bài báo cáo này, người viết đã nghiên cứu về kiến trúc và cách sử dụng của công nghệ Google Bigtable. Đồng thời đã hiện thực một sản phẩm demo chạy trên một dữ liệu thực tiễn, cho thấy tiềm năng ứng dụng của công nghệ Bigtable trong thực tiễn, bên cạnh các giải pháp khác như MongoDB, Cassandra hay Neo4j.

Tài liệu tham khảo

[1] Macak, Martin Stovcik, Matus Buhnova, Barbora. (2020). The Suitability of Graph Databases for Big Data Analysis: A Benchmark. 213-220. 10.5220/0009350902130220.

[2] https://www.oracle.com/big-data/what-is-graph-database/

[3] R. Jain, S. Iyengar and A. Arora, "Overview of popular graph databases," 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013, pp. 1-6, doi: 10.1109/ICCCNT.2013.6850236.

[4] Rania Soussi, "Graph Database For collaborative Communities".

[5] Mike Buerli, "The Current State of Graph Databases".

[6] Fay Chang and Jeffrey Dean and Sanjay Ghemawat and Wilson C. Hsieh and Deborah A. Wallach and Mike Burrows and Tushar Chandra and Andrew Fikes and Robert E. Gruber, "Bigtable: A Distributed Storage System for Structured Data".

[7] https://www.kaggle.com/stoney71/new-york-city-transport-statistics

A

Kết quả hiện thực

A.1 Thực hiện tìm kiếm vị trí của một phương tiện

Tìm kiếm vị trí của một phương tiện cụ thể là câu truy vấn đơn giản nhất. Mục tiêu của câu truy vấn này là truy xuất thông tin vị trí của phương tiện mã số *NYCT_5824* trong một tiếng đồng hồ từ 12:00AM đến 1:00AM ngày 01/06/2017. Với các dữ kiện trên, kết hợp với thiết kế row key trong nội dung trình bày chính, khóa hàng dữ liệu cần được truy vấn sẽ là *MTA/M86-SBS/1496275200000/NYCT_5824*

Đoạn mã nguồn được sử dụng:

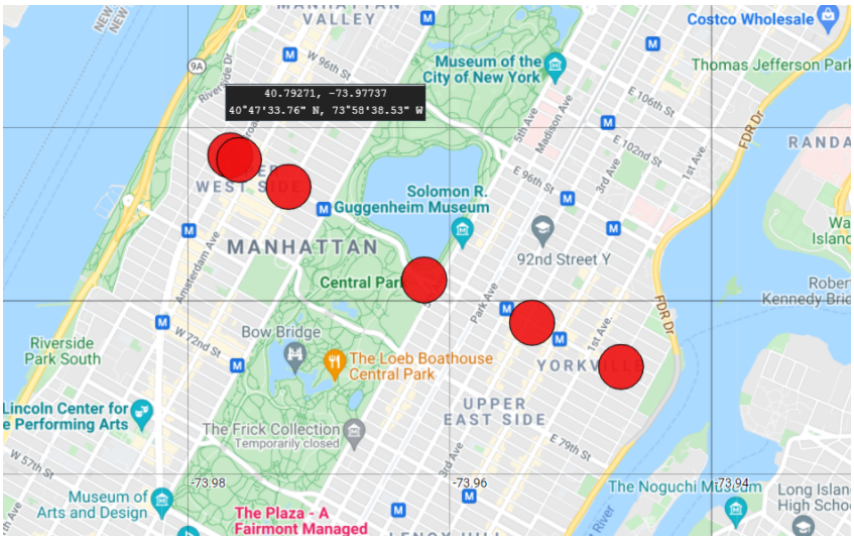
```
private static final byte[] COLUMN_FAMILY_NAME = Bytes.toBytes("cf");
private static final byte[] LAT_COLUMN_NAME = Bytes.toBytes("VehicleLocation.Latitude"
private static final byte[] LONG_COLUMN_NAME = Bytes.toBytes("VehicleLocation.Longitu
```

```
String rowKey = "MTA/M86-SBS/1496275200000/NYCT_5824";
Result getResult =
    table.get(
        new Get(Bytes.toBytes(rowKey))
            .addColumn(COLUMN_FAMILY_NAME, LAT_COLUMN_NAME)
            .addColumn(COLUMN_FAMILY_NAME, LONG_COLUMN_NAME));
```

Kết quả sau khi chạy trên node dữ liệu của Google sẽ có định dạng như sau:

```
Lookup a specific vehicle on the M86 route on June 1, 2017 from 12:00am to 1:00am:
40.781212,-73.961942
40.776163,-73.946949
40.778714,-73.953731
40.786553,-73.972333
40.788416,-73.976748
40.788136,-73.976083
```

Sử dụng công cụ MapMarker App (https://maps.co/gis/) để hiển thị các tọa độ thu được trên bản đồ thật sẽ thu được kết quả:



A.2 Thực hiện quét toàn bộ vị trí của tất cả các tuyến

Tác vụ này là tác vụ mang lại nhiều ý nghĩa về mặt thống kê nhất và thường được sử dụng trong thực tiễn. Việc trực quan hóa dữ liệu vị trí của toàn bộ tuyến xe buýt trong một khung thời gian có thể giúp người quản lý đánh giá được hiệu quả của các tuyến xe này, đồng thời nhận định được những điểm nóng trong giao thông cần phải cải thiện.

Nhờ vào việc thiết kế hợp lý, ta thấy rằng dữ liệu của các tuyến xe khác nhau trong cùng một khoảng thời gian sẽ có xu hướng nằm gần nhau, do đó việc truy vấn rất hiệu quả. Đoạn mã nguồn được sử dụng:

```
private static final String[] MANHATTAN_BUS_LINES = {"M1","M2","M3",...

Scan scan;
ResultScanner scanner;
List<RowRange> ranges = new ArrayList<>();
for (String busLine : MANHATTAN_BUS_LINES) {
  ranges.add(
      new RowRange(
          Bytes.toBytes("MTA/" + busLine + "/1496275200000"), true,
          Bytes.toBytes("MTA/" + busLine + "/1496275200001"), false));
}
Filter filter = new MultiRowRangeFilter(ranges);
scan = new Scan();
scan.setFilter(filter);
scan.setMaxVersions(Integer.MAX_VALUE)
    .addColumn(COLUMN_FAMILY_NAME, LAT_COLUMN_NAME)
    .addColumn(COLUMN_FAMILY_NAME, LONG_COLUMN_NAME);
```

```
scan.withStartRow(Bytes.toBytes("MTA/M")).setRowPrefixFilter(Bytes.toBytes("MTA/M"));
scanner = table.getScanner(scan);
```

Kết quả được trực quan hóa dưới dạng biểu đồ nhiệt, các điểm màu đỏ cho thấy tần suất xuất hiện nhiều của các phương tiện.