ISSN 2354-0575

SO SÁNH CƠ SỞ DỮ LIỆU QUAN HỆ VÀ CƠ SỞ DỮ LIỆU ĐỒ THỊ TRONG QUẢN LÝ DỮ LIỆU INTERNET KẾT NỐI VẠN VẬT

Nguyễn Hữu Đông, Vũ Huy Thế, Nguyễn Văn Quyết*
Trường Đại học Sư phạm Kỹ thuật Hưng Yên

Ngày tòa soạn nhận được bài báo: 05/03/2020
Ngày phản biện đánh giá và sửa chữa: 25/05/2020
Ngày bài báo được duyệt đăng: 15/06/2020

Tóm tắt:

Trong môi trường Internet kết nối vạn vật (Internet of Thing - IoT), các thực thể với nhiều thuộc tính và số lượng khác nhau được kết nối tạo thành một mạng lưới dày đặc. Ở đó, không chỉ máy tính và các thiết bị điện tử mà cả các thực thể khác như con người, vị trí và các ứng dụng cũng kết nối với nhau. Việc hiểu và quản lý các kết nối này đóng vai trò quan trọng cho việc phát triển các dịch vụ IoT mới trong kinh doanh. Để giải quyết vấn đề này, các cách tiếp cận truyền thống sử dụng các hệ quản trị cơ sở dữ liệu quan hệ như MySQL hay MSSQL để lưu trữ và truy vấn dữ liệu IoT. Tuy nhiên, sử dụng cơ sở dữ liệu quan hệ sẽ không linh hoạt và hiệu quả khi phải xử lý các dữ liệu kết nối hỗn hợp trong IoT bởi vì nhữ dữ liệu này có mối liên quan phức tạp theo chiều sâu, đòi hỏi các câu truy vấn lồng và các phép toán nối (JOIN) phức tạp trên nhiều bảng dữ liệu. Gần đây, cơ sở dữ liệu đồ thị đã được phát triển để lưu trữ và phân tích các dữ liệu có tính kết nối dày đặc. Trong bài báo này, chúng tôi phân tích và so sánh toàn diện giữa cơ sở dữ liệu quan hệ và cơ sở dữ liệu đồ thị cho việc quản lý dữ liệu Internet kết nối vạn vật. Thông qua việc so sánh trên nhiều khía cạnh và các kết quả thực nghiệm, chúng tôi chỉ ra rằng cơ sở dữ liệu đồ thị rất phù hợp cho việc lưu trữ và phân tích dữ liệu Internet kết nối vạn vật.

Từ khóa: *Graph Database, Graph Queries, Query Performance, Connected Data, IoT Data Management*

1. Giới thiệu

Trong những năm gần đây, các sản phẩm và dịch vụ Internet kết nối vạn vật đang được sử dụng rất nhiều trong cuộc sống của chúng ta [1][2]. Chúng không chỉ giúp cho cuộc sống trở nên an toàn, thuận tiện, mà còn cải thiện hiệu năng công việc cũng như tạo thêm giá trị trong kinh doanh. Ví dụ, trong một tòa nhà thông minh lớn có đến hàng vài chục nghìn cảm biến, thiết bị IoT kết nối với nhau để cung cấp các thông tin cho hệ thống quản lý thông tin tòa nhà [3]. Những thông tin này sau khi được xử lý sẽ hỗ trợ hệ thống ra quyết định trong những tình huống cần thiết như cứu hộ khẩn cấp [4]. Một ví dụ khác, trong một trang trại thông minh được trang bị rất nhiều cảm biến về nhiệt độ, độ ẩm, các thiết bị khác như camera để theo dõi sâu bệnh, sự phát triển của cây trồng…dữ liệu từ các thiết bị này có thể thu thập và biểu diễn thành tri thức, lưu trữ, và khai thác để hỗ trợ các hệ thống ra quyết định trong trang trại thông minh [2]. Việc lựa chọn một cơ sở dữ liệu hỗ trợ lưu trữ và khai phá tốt nhất dữ liệu kết nối là một thực tế hiển nhiên cho việc quản lý dữ liệu IoT.

Theo cách tiếp cận truyền thống, một số hệ thống phần mềm IoT sử dụng cơ sở dữ liệu quan hệ như MySQL, MSSQL, MariaDB để lưu trữ và truy vấn dữ liệu IoT. Tuy nhiên, sử dụng cơ sở dữ liệu quan hệ là không đủ mạnh để quản lý dữ liệu kết nối sâu trong môi trường IoT, ở đó, dữ liệu tồn tại ở dạng hỗn hợp cả có cấu trúc và không có cấu trúc. Việc truy vấn dữ liệu kết nối sâu đòi hỏi rất nhiều câu truy vấn lồng cũng như sử dụng rất nhiều phép toán JOIN từ nhiều bảng dữ liệu. Điều này khiến cho thời gian xử lý truy vấn tăng lên, hệ thống khó đáp dứng được việc xử lý thời gian thực theo như yêu cầu của nhiều hệ thống IoT.

Gần đây, các cơ sở dữ liệu phi quan hệ (non-relation hay NoSQL) đã được nhiều nhóm nghiên cứu, doanh nghiệp quan tâm và phát triển như khóa-giá trị (key-value), (họ cột) column-family, tài liệu (document), và cơ sở dữ liệu đồ thị (graph database) [5]. Trong số những cơ sở dữ liệu NoSQL trên, cơ sở dữ liệu đồ thị là một trong những cơ sở dữ liệu phổ biến nhất được sử dụng bởi nhiều doanh nghiệp.

Trong bài báo này, chúng tôi trình bày một so

sánh toàn diện giữa cơ sở dữ liệu quan hệ và cơ sở dữ liệu đồ thị cho việc quản lý dữ liệu Internet kết nối vạn vật. Đầu tiên, chúng tôi trình bày các đặc tính của dữ liệu IoT cũng như các thách thức trong việc quản lý những dữ liệu này. Sau đó chúng tôi so sánh cơ sở dữ liệu liệu quan hệ và cơ sở dữ liệu đồ thị trên nhiều khía cạnh bao gồm: mô hình dữ liệu, hiệu năng truy vấn, hỗ trợ giao dịch, và tính mở rộng. Cuối cùng, chúng tôi đánh giá hiệu năng của việc truy vấn dữ liệu của cơ sở dữ liệu quan hệ và cơ sở dữ liệu đồ thị sử dụng các dữ liệu thực tế và dữ liệu tổng hợp. Thông qua các kết quả thực nghiệm, chúng tôi chỉ ra rằng cơ sở dữ liệu đồ thị rất phù hợp cho việc lưu trữ và phân tích dữ liệu Internet kết nối vạn vật.

2. Các đặc tính của dữ liệu IoT và các thách thức trong quản lý dữ liệu IoT

Để thể hiện rõ sự cần thiết của một cơ sở dữ liệu mới thay cho cơ sở dữ liệu quan hệ trong việc quản lý và truy vấn dữ liệu kết nối trong môi trường IoT, trong phần này, chúng tôi trình bày các đặc tính của dữ liệu IoT cũng như thách thức của việc quản lý dữ liệu IoT.

2.1. Tính hỗn hợp

Với nhiều thực thể khác nhau trong môi trường IoT, các hệ thống IoT sẽ tạo ra các kiểu dữ liệu khác nhau bao gồm cả dữ liệu có cấu trúc, bán cấu trúc, hay không có cấu trúc [6]. Ví dụ, trong một hệ thống IoT dành cho quản lý tòa nhà thông minh, dữ liệu sinh ra từ rất nhiều các thiết bị như cảm biến nhiệt độ, khói, .. là không có cấu trúc; trong khi, thông tin về người hay phòng trong các tòa nhà đó có thể được quản ý dưới dạng các bảng dữ liệu có cấu trúc hoặc các tệp tin bán cấu trúc XML [7][8]. Do đó, việc quản lý dữ liệu IoT hỗn hợp sao cho chúng có thể khai thác một cách dễ dạng trong các hệ thống IoT được xem như là một thách thức.

2.2. Tính kế nối dày đặc

Trong một môi trường IoT có rất nhiều loại thực thể, mỗi loại thực thể tồn tại với số lượng và thuộc tính khác nhau [9]. Chúng kết nối với nhau tạo thành mạng lưới dày đặc. Ví dụ, trọng một tòa nhà thông minh tên là Edge tại Amsterdam có khoảng 22,000 thiết bị IoT kết nối với nhau [3]. Một vài câu hỏi có thể đặt ra ở đây khi một cảm biến ở một phòng chỉ báo nhiệt độ phòng đang lạnh hơn thông thường như "tại sao nhiệt độ phòng lại lạnh hơn? trạng thái của hệ thống sưởi trong phòng đấy như thế nào?, hay có bao nhiêu người đang ở trong phòng?". Để trả lời những câu hỏi đó, chúng ta cần phải phân tích được những thứ xung quang liên quan đến căn phòng. Bởi vậy, việc hiểu và quản lý các kết nối giữa các thực thể trong môi trường IoT là một thách thức và đóng vai trò quan trọng trong các hệ thống IoT.

2.3. Tính thay đổi động

Hầu hết các ứng dụng IoT làm việc trong môi trường dữ liệu thay đổi nhanh do các thực thể trong liên tục được thêm/bớt vào hệ thống làm cho kết nối giữa các thực thể cũng liên tục thay đổi [10][11]. Ví dụ, trong dịch vụ giao thông thông minh, một chiếc ô tô có thể chia sẻ thông tin trạng thái của tuyến đường nó đang đi với các xe khác xung quanh nó. Các kết nối giữa các xe có thể tạo thành một mạng lưới hay một đồ thị. Nhưng, các xe này di chuyển nhanh và có thể nhanh chóng ngắt kết nối với các xe khác, vậy một thách thức ở đây là làm thế nào để có thể quản lý những kết nối này một cách hiệu quả. Do đó, nó đòi hỏi một mô hình dữ liệu dễ dạng biểu diễn thông tin các thực thể, hỗ trợ cập nhật mối quan hệ giữa các thực thế mà không ảnh hưởng đến tính hoạt động của các hệ thống IoT.

2.4. Dữ liệu thời gian thực lớn

Dữ liệu lớn được tạo ra từ hàng nghìn loại thiết bị và dịch vụ như cảm biến, camera, hay mạng xã hội liên quan đến một hệ thống IoT [12][13]. Ví dụ, một lượng lớn ảnh hay video được sinh ra theo thời gian thực qua việc sử dụng các thiết bị camera an ninh trong tòa nhà, nó không phù hợp để lưu trữ trong một cơ sở dữ liệu quan hệ thông thường như MySQL hay MSSQL. Do vậy, việc mô hình hóa dữ liệu sao cho nó dễ dàng trong xử lý thời gian thực là một thách thức lớn đối với các hệ thống IoT.

3. So sánh cơ sở dữ liệu quan hệ và cơ sở diệu đồ thị

Để đánh giá sự khác biệt giữa cơ sở dữ liệu quan hệ và cơ sở dữ liệu đồ thị, chúng tôi thực hiện việc so sánh trên bốn đặc tính quan trọng bao gồm: mô hình dữ liệu, hiệu năng truy vấn, hỗ trợ giao dịch, và khả năng mở rộng.

3.1. Mô hình dữ liệu

Cơ sở dữ liệu quan hệ sử dụng cấu trúc cố định

với các bảng được xác định trước bởi các hàng và các cột. Điều này làm cho chúng không phù hợp để lưu trữ dữ liệu IoT không có cấu trúc hoặc dữ bán cấu trúc. Trong khi đó, cơ sở dữ liệu đồ thị sử dụng cấu trúc linh hoạt bằng cách sử dụng cấu trúc đồ thị, trong đó, các nút được sử dụng để thể hiện thông tin các thực thể và các cạnh thể hiện mối quan hệ giữa các thực thể. Nhờ đó, nó dễ dàng mô tả dữ liệu không đồng nhất và dữ liệu kết nối dày đặc. Cơ sở dữ liệu quan hệ mô tả mối quan hệ giữa các thực thể bằng sử dụng các mối quan hệ tiêu chuẩn: một-một, một-nhiều và nhiều-nhiều. Các bảng liên kết với nhau sử dụng các khóa ngoại để đảm bảo tính thống nhất của dữ liệu. Điều này gây ra khó khăn trong việc lưu trữ và xử lý dữ liệu liên tục thay đổi trong môi trường IoT. Với cơ sở dữ liệu đồ thị, việc thêm và xóa các thực thể và quan hệ của chúng là những thao tác đơn giản.

3.2. Hiệu năng truy vấn

Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL) để truy cập dữ liệu. Ngôn ngữ SQL được định nghĩa rất rõ ràng và sử dụng phổ biến trong cả học thuật và các hệ thống trong doanh nghiệp. Tuy nhiên, cơ sở dữ liệu quan hệ không được thiết kế để xử lý dữ liệu lớn và dữ liệu có tính kết nối dày đặc như dữ liệu IoT trong các ứng dụng hiện đại. Do đó, hiệu năng truy vấn có thể thấp do phải sử dụng nhiều câu truy vấn lồng nhau hoặc một số lượng lớn các phép nối từ nhiều bảng. Ngược lại, đồ thị cơ sở dữ liệu được xây dựng có chủ đích để lưu trữ và xử lý dữ liệu được kết nối dày đặc; do đó, nó có thể đạt được hiệu năng cao trong việc truy vấn dữ liệu IoT. Điểm chính giúp cơ sở dữ liệu đồ thị đạt được hiệu năng cao đó là việc áp dụng các kỹ thuật duyệt đồ thị như BFS và DFS. Trong khi đó, cơ sở dữ liệu quan hệ sử dụng các kỹ thuật kết hợp quét và băm dữ liệu để so sánh dẫn đến tốn kém chi phí khi thao tác với các bảng lớn hoặc nhiều phép nối. Do đó, hiệu năng truy vấn cơ sở dữ liệu quan hệ giảm khi tăng số lượng bản ghi trong bảng và số lượng mối quan hệ giữa các bảng, trong khi với cơ sở dữ liệu đồ thị, hiệu năng của nó chỉ giảm khi tăng số lượng kết nối giữa các thực thể.

3.3. Hỗ trợ giao dịch

Một trong những chức năng quan trọng của cơ sở dữ liệu quan hệ khiến chúng là lựa chọn ưu tiên trong các doanh nghiệp phần mềm là ứng dụng là ACID (Atomicity - nguyên tử, Consistency - nhất quán, Isolation - độc lập, Durability - bền vững). Các thuộc tính ACID cung cấp một cơ chế để đảm bảo dữ liệu của giao dịch không bị hỏng do bất kỳ lý do nào (ví dụ: giao dịch chuyển tiền giữa các tài khoản trong ngân hàng). Trong khi hầu hết các cơ sở dữ liệu NoSQL sử dụng mô hình nhất quán BASE (Tính khả dụng cơ bản, trạng thái mềm, tính nhất quán cuối cùng) để hỗ trợ các giao dịch trong cơ sở dữ liệu, cơ sở dữ liệu đồ thị hiện tại (ví dụ: Neo4J, OrientDB) giữ lại các thuộc tính ACID được yêu cầu bởi các ứng dụng IoT hiện đại.

3.4. Khả năng mở rộng

Để xử lý dữ liệu lớn, khả năng mở rộng là rất quan trọng trong các hệ thống IoT. Cơ sở dữ liệu quan hệ sử dụng khả năng mở rộng theo chiều dọc, điều đó có nghĩa là việc cải thiện hiệu năng xử lý dữ liệu lớn được thực hiện bằng cách nâng cấp dung lượng lưu trữ và khả năng tính toán (ví dụ: sử dụng ổ SSD, tăng số lõi CPU, v.v.) phần cứng hiện có trong hệ thống. Mở rộng theo chiều dọc thường tốn kém chi phí và khả năng phục hồi khi có lỗi hệ thống không được đảm bảo khi lỗi máy chủ cơ sở dữ liệu. Trong khi đó, cơ sở dữ liệu đồ thị sử dụng khả năng mở rộng theo chiều ngang, có nghĩa là khi lượng dữ liệu IoT tăng nhanh, chúng ta thêm nhiều tài nguyên hơn (ví dụ: tăng số lượng máy chủ) vào hệ thống để mở rộng lưu trữ và cải thiện hiệu năng truy vấn.

4. Kết quả thực nghiệm

Trong phần này, chúng tôi tạo ra các thực nghiệm để so sánh cơ sở dữ liệu quan hệ và cơ sở dữ liệu đồ thị về hiệu năng truy vấn. Để làm điều này, chúng tôi sử dụng hai bộ dữ liệu bao gồm: Sakila và Gnutella.

- Sakila: bộ dữ liệu thực tế của cửa hàng cho thuê DVD được cung cấp bởi nhóm phát triển MySQL [14]. Bộ dữ liệu này có 16 bảng và 47,271 bản ghi. Chúng tôi thực hiện chuyển đổi sang nhập vào cơ sở dữ liệu đồ thị (Neo4J) với 40,810 nút và 114,706 cạnh.
- Gnutella: bộ dữ liệu thực tế của mạng ngang hàng Internet [15]. Chúng tôi nhập bộ dữ liệu này vào một bảng trong MySQL với 138,142 bản ghi. Bộ dữ liệu cũng được nhập vào Neo4J với 60,000 nút và 138,142 cạnh.

Bảng 1. Hiệu năng truy vấn của SQL trên dữ liệu Sakila

Kiểu truy vấn	#Câu truy vấn	Thực thi lần 1 (ms)	Thực thi lần 2 (ms)	Thực thi lần 3 (ms)	Thời gian trung bình (ms)	Độ lệch chuẩn (ms)
Tra cứu (Look Up)	Q1	15	15	16	15.33	0.58
	Q2	16	15	15	15.33	0.58
	Q3	16	15	15	15.33	0.58
Phạm vi (Range)	Q4	16	16	16	16.00	0.00
	Q5	16	16	15	15.67	0.58
	Q6	16	15	16	15.67	0.58
Phức tạp (Complex)	Q7	31	31	32	31.33	0.58
	Q8	63	62	62	62.33	0.58
	Q9	94	93	79	88.67	8.39
Tập hợp (Aggregation)	Q10	32	32	31	31.67	0.58
	Q11	79	78	78	78.33	0.58
	Q12	94	94	93	93.67	0.58

Bảng 2. Hiệu năng truy vấn bằng Cypher trên dữ liệu Sakila

Kiểu truy vấn	#Câu truy vấn	Thực thi lần 1 (ms)	Thực thi lần 2 (ms)	Thực thi lần 3 (ms)	Thời gian trung bình (ms)	Độ lệch chuẩn (ms)
Tra cứu (Look Up)	Q1	1	1	1	1.00	0.00
	Q2	1	1	1	1.00	0.00
	Q3	1	1	1	1.00	0.00
Phạm vi (Range)	Q4	1	1	1	1.00	0.00
	Q5	1	1	1	1.00	0.00
	Q6	1	1	1	1.00	0.00
Phức tạp (Complex)	Q7	3	3	2	2.67	0.58
	Q8	21	21	19	20.33	1.15
	Q9	10	10	10	10.00	0.00
Tập hợp (Aggregation)	Q10	27	23	21	23.67	3.06
	Q11	33	31	31	31.67	1.15
	Q12	77	77	75	76.33	1.15

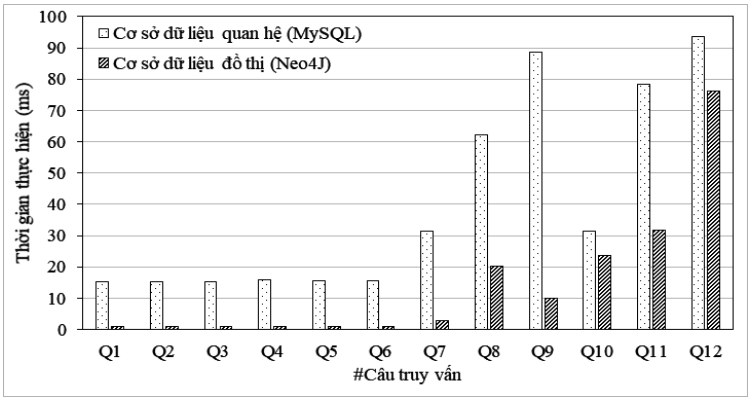
Bảng 3. Hiệu năng truy vấn bằng SQL trên dữ liệu Gnutella

Kiểu truy vấn	#Câu truy vấn	Thực thi lần 1 (ms)	Thực thi lần 2 (ms)	Thực thi lần 3 (ms)	Thời gian trung bình (ms)	Độ lệch chuẩn (ms)
Tra cứu (Look Up)	Q1	47	47	47	47.00	0.00
	Q2	47	46	47	46.67	0.58
	Q3	47	47	47	47.00	0.00
Phạm vi (Range)	Q4	62	62	62	62.00	0.00
	Q5	63	63	62	62.67	0.58
	Q6	63	63	63	63.00	0.00
Phức tạp (Complex)	Q7	157	156	156	156.33	0.58
	Q8	406	391	390	395.67	8.96
	Q9	890	875	875	880.00	8.66
Tập hợp (Aggregation)	Q10	109	94	94	99.00	8.66
	Q11	141	141	125	135.67	9.24
	Q12	125	110	109	114.67	8.96

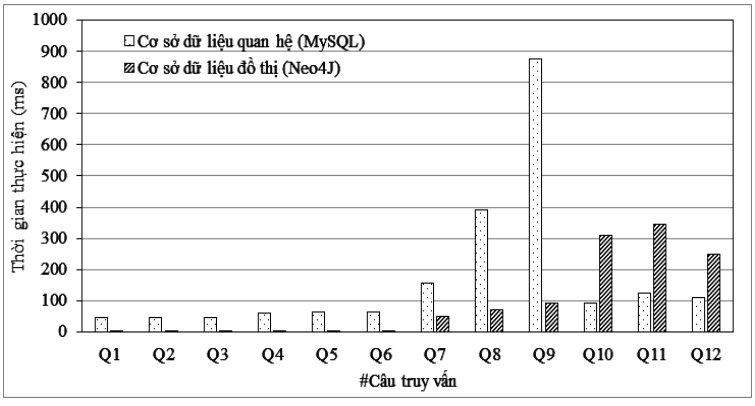
Bảng 4. Hiệu năng truy vấn bằng Cypher trên dữ liệu Gnutella

Kiểu truy vấn	#Câu truy vấn	Thực thi lần 1 (ms)	Thực thi lần 2 (ms)	Thực thi lần 3 (ms)	Thời gian trung bình (ms)	Độ lệch chuẩn (ms)
Tra cứu (Look Up)	Q1	1	1	1	1.00	0.00
	Q2	1	1	1	1.00	0.00
	Q3	1	1	1	1.00	0.00
Phạm vi (Range)	Q4	1	1	1	1.00	0.00
	Q5	1	1	1	1.00	0.00
	Q6	1	1	1	1.00	0.00
Phức tạp (Complex)	Q7	55	55	49	53.00	3.46
	Q8	71	71	71	71.00	0.00
	Q9	95	95	94	94.67	0.58
Tập hợp (Aggregation)	Q10	350	316	309	325.00	21.93
	Q11	385	359	345	363.00	20.30
	Q12	264	254	248	255.33	8.08

Chúng tôi đánh giá bốn loại truy vấn phổ biến bao gồm Tra cứu (Look up), Phạm vi (Range), Phức tạp (JOIN / NESTED), *và* Tập hợp (Aggregation) thường được sử dụng để trích xuất các tri thức từ dữ liệu IoT. *Đối với mỗi tập dữ liệu, chúng tôi* viết ra mười hai truy vấn, mỗi loại truy vấn gồm ba *câu* truy vấn. Các truy vấn được viết bằng cả ngôn ngữ SQL để chạy trên MySQL và ngôn ngữ Cypher để chạy trên Neo4J. Các kết quả thực nghiệm được minh họa từ Bảng 1 đến Bảng 4.



Hình 1. So sánh hiệu năng giữa cơ sở dữ liệu quan hệ và cơ sở dữ liệu đồ thị trên bộ dữ liệu Sakila



Hình 2. So sánh hiệu năng giữa cơ sở dữ liệu quan hệ và cơ sở dữ liệu đồ thị trên bộ dữ liệu Gnutella

So sánh hiệu năng truy vấn giữa cơ sở dữ liệu quan hệ và biểu đồ cơ sở dữ liệu trên tập dữ liệu Sakila và Gnutella được mô tả trong Hình 1 và Hình 2. Từ kết quả, chúng tôi thấy rằng sử dụng truy vấn Cypher trên Neo4J có được hiệu năng tốt hơn so với việc sử dụng các truy vấn SQL trên MySQL trong tất cả các trường hợp nói chung. Cụ thể, các câu lệnh tra cứu và truy vấn phạm vi có chi phí thấp trên cả cơ sở dữ liệu quan hệ và cơ sở dữ liệu đồ thị. Trong trường hợp kiểm tra các truy vấn phức tạp tên dữ liệu Sakila hoặc truy vấn lồng nhau trên dữ liệu Gnutella, hiệu năng sử dụng truy vấn Cypher trên cơ sở dữ liệu đồ thị là nhiều nhanh hơn so với sử dụng truy vấn SQL trên cơ sở dữ liệu quan hệ. Chúng tôi quan sát thấy truy vấn với Cypher giảm thời gian thực hiện trung bình khoảng 12, 3, 9 lần so với truy vấn SQL trong trường hợp bộ dữ liệu Sakila và giảm 3, 5, và 9 lần với bộ dữ liệu Gnutella tương ứng các câu truy vấn #Q7, # Q8, và # Q9. Chúng tôi cũng quan sát thấy các truy vấn tập hợp trên cơ sở dữ liệu đồ thị thường tốn nhiều thời gian thực thi hơn. Thật vậy, hiệu năng của chúng xấp xỉ bằng với các truy vấn SQL (#10, #11, #12) trên tập dữ liệu Sakila, thậm chí chậm hơn nhiều lần trong trường hợp thử nghiệm với bộ dữ liệu Gnutella.

5. Kết luận

Bài báo này phân tích và so sánh toàn diện giữa cơ sở dữ liệu quan hệ và cơ sở dữ liệu đồ thị trong việc quản lý dữ liệu Internet kết nối vạn vật. Chúng tôi đã so sánh trên các đặc tính quan trọng của các cơ sở dữ liệu gồm: mô hình dữ liệu, hiệu năng truy vấn, hỗ trợ giao dịch, và khả năng mở rộng. Chúng tôi cũng đánh giá và so sánh trên thực nghiệm về mặt hiệu năng truy vấn của các cơ sở dữ liệu với hai bộ dữ liệu thực tế là Sakila và Gnutella. Thông qua việc so sánh trên nhiều đặc tính và kết quả thực nghiệm chúng tôi chỉ ra rằng cơ sở dữ liệu đồ thị là phù hợp hơn trong việc lưu trữ và phân tích dữ liệu kết nối trong môi trường IoT.

Tài liệu tham khảo

[1]. Lee, I. and Lee, K., 2015. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, **58(4)**, pp.431-440.

[2]. Van-Quyet, Nguyen, et al. “Design of a Platform for Collecting and Analyzing Agricultural Big Data.” *JDCS* **vol. 18, no.1**, pp. 149-158, 2017.

[3]. A. Mulholland, “Iot: Where do graphs fit with business requirements.” [Online]. Available: https://neo4j.com/blog/iot-graphs-business-requirements/ (Accessed June 28, 2020).

[4]. Lin, C.Y., Chu, E., Ku, L.W. and Liu, J., Active disaster response system for a smart building. *Sensors*, **14(9)**, pp.17451-17470, 2014. .

[5]. Van-Quyet Nguyen, Huu-Duy Nguyen, Giang-Truong Nguyen, Kyungbaek Kim, “A Graph Model of Heterogeneous IoT Data Representation: A Case Study from Smart Campus Management”, In *Proceedings of KISM Fall Conference* 2018.

[6]. S. Wu, L. Bao, Z. Zhu, F. Yi, and W. Chen, “Storage and retrieval of massive heterogeneous iot data based on hybrid storage,” in *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. *IEEE*, pp. 2982–2987, 2017.

[7]. S. K. Sowe, T. Kimata, M. Dong, and K. Zettsu, “Managing heterogeneous sensor data on a big data platform: Iot services for data-intensive science,” in *2014 IEEE 38th International Computer Software and Applications Conference Workshops. IEEE*, pp. 295–300, 2014.

[8]. F. Ullah, M. A. Habib, M. Farhan, S. Khalid, M. Y. Durrani, and S. Jabbar, “Semantic interoperability for big-data in heterogeneous iot infrastructure for healthcare,” *Sustainable cities and society*, **vol. 34**, pp. 90–96, 2017.

[9]. Arora, Vaibhav, Faisal Nawab, Divyakant Agrawal, and Amr El Abbadi. “Multi-representation based data processing architecture for IoT applications.” In *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*. IEEE, pp. 2234-2239, 2017.

[10]. D. Puschmann, P. Barnaghi, and R. Tafazolli, "Adaptive clustering for dynamic iot data streams," *IEEE Internet of Things Journal*, **vol. 4, no. 1**, pp. 64–74, 2017.
[11]. M. Bermudez-Edo, T. Elsaleh, P. Barnaghi, and K. Taylor, "Iot-lite: a lightweight semantic model for the internet of things and its use with dynamic semantics," *Personal and Ubiquitous Computing*, **vol. 21, no. 3**, pp. 475–487, 2017.
[12]. K. Yasumoto, H. Yamaguchi, and H. Shigeno, "Survey of real-time processing technologies of iot data streams," *Journal of Information Processing*, **vol. 24, no. 2,** pp. 195–202, 2016.
[13]. S. Verma, Y. Kawamoto, Z. M. Fadlullah, H. Nishiyama, and N. Kato, "A survey on network methodologies for real-time analytics of massive iot data and open research issues," *IEEE Communications Surveys & Tutorials*, **vol. 19, no. 3**, pp. 1457–1477, 2017.
[14]. O. Corporation, "Sakila sample database." [Online]. Available: https://dev.mysql.com/doc/sakila/ (Accessed June 28, 2020).
[15]. M. Ripeanu and I. Foster and A. Iamnitchi. "Mapping the Gnutella Network: Properties of Large-Scale Peer-toPeer Systems and Implications for System Design". *IEEE Internet Computing Journal,* 2002.

A COMPREHENSIVE COMPARISON OF RELATIONAL DATABASES AND GRAPH DATABASES FOR HETEROGENEOUS IOT DATA MANAGEMENT

Abstract:

In an Internet of Thing (IoT) environment, entities with different attributes and capacities are going to be collaborated in a highly connected. Specifically, not only the mechanical and electronic devices but also other entities such as people, locations and applications are connected to each other. Understanding and managing these connections play an important role for businesses, which identify opportunities for new IoT services. Traditional approaches for storing and querying IoT data are used of relational database management systems (RDMS) such as MySQL or MSSQL. However, using RDMS is not flexible and suffcient for handling highly connected heterogeneous IoT data because these data have deeply complex relationships which require nested queries and complex joins on multiple tables. Recently, graph databases have been recently developed for storing and analyzing highly connected data. This paper presents an analysis and a comprehensive comparison of relational databases and graph databases for heterougeneous IoT data management. Through the comparison in various aspects and experimental results, we find that graph databases are applicable for storing and analyzing the IoT connected data.

Keywords: *Graph Database, Graph Queries, Query Performance, Connected Data, IoT Data Management.*