

ĐẠI HỌC QUỐC GIA THÀNH PHỐ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
Ngành Kỹ Thuật Máy Tính

**Ứng dụng Internet of Things để
phát triển hệ thống thu thập và
giám sát khí thải của phương tiện
giao thông**

Giảng viên hướng dẫn:

TS. Phạm Hoàng Anh

Sinh viên thực hiện:

5120000 - Nguyễn Tân Tùng

51202655 - Huỳnh Phạm So Ny

51200000 - Nguyễn Mạnh Cường

Tháng 12 Năm 2016

Lời cam đoan

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Tháng 12 Năm 2016

Lời cảm ơn

Em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đối với các thầy cô của trường Đại học Bách Khoa thành phố Hồ Chí Minh, đặc biệt là các thầy cô khoa Khoa Học và Kỹ Thuật Máy Tính của trường đã tạo điều kiện cho em thực tập ở khoa để có nhiều thời gian cho luận văn tốt nghiệp. Đặc biệt, xin gửi lời cảm ơn chân thành Tiến sĩ Phạm Hoàng Anh đã nhiệt tình hướng dẫn hướng dẫn hoàn thành tốt đợt thực tập. Cám ơn các bạn Nguyễn Thị Trang 91103730, Đỗ Thành Nhân G1302688 đã có những đóng góp quý báu cùng với nhóm.

Trong quá trình thực tập, cũng như là trong quá trình làm bài báo cáo, khó tránh khỏi sai sót, rất mong các Thầy, Cô bỏ qua. Đồng thời do trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên bài báo cáo không thể tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp Thầy, Cô để em học thêm được nhiều kinh nghiệm và sẽ hoàn thành tốt hơn bài luận văn sắp tới. Em xin chân thành cảm ơn!

Nhóm thực hiện đề tài.

Tóm tắt

Do nhu cầu áp dụng khoa học và kỹ thuật vào phát triển hệ thống giám sát môi trường và giao thông, sinh viên nhóm chúng tôi đã tìm hiểu và thực hiện dự án hệ thống thu thập tình trạng ô nhiễm do khí thải của các phương tiện giao thông. Nhờ sự phát triển mạnh mẽ của Internet of Things, đề tài được định hướng phát triển theo mô hình IoT kết hợp với các kiến thức về môi trường nhằm để xây dựng hệ thống thiết bị đo đặc và lấy dữ liệu để phân tích. Nhóm bắt đầu tìm hiểu về các loại khí thải của phương tiện giao thông, tìm hiểu các cảm biến, các loại vi xử lý hỗ trợ tốt cho hệ thống IoT, kết hợp với hệ thống Cloud Database và ứng dụng Web theo dõi tình trạng ô nhiễm thực tế trên một số đoạn đường đã được đo đặc.

Mục lục

Danh sách hình vẽ	vii
Danh sách bảng	viii
1 Giới thiệu	1
1.1 Ý tưởng và tính cấp thiết của đề tài	1
1.2 Mục tiêu và nhiệm vụ của đề tài	2
1.2.1 Mục tiêu đề tài	2
1.2.2 Nhiệm vụ đề tài	3
2 IoT và Các kiến thức cơ bản về IoT	4
2.1 Giới thiệu về Internet of Things và các ứng dụng giám sát	4
2.1.1 Internet of Things (IoT)	4
2.1.2 Các hệ thống giám sát được phát triển dựa trên IoT	12
2.2 Giới thiệu về các công cụ hỗ trợ phát triển ứng dụng IoT	17
2.3 Những yếu tố ảnh hưởng môi trường từ khí thải phương tiện giao thông	17
2.4 Các hệ thống quan trắc hiện hữu	19
2.5 Các thiết bị phần cứng	22
2.5.1 Thiết bị thu thập dữ liệu	22
2.5.2 Các loại sensor	23
2.5.3 Mạch vi xử lý Arduino	23
2.5.4 Module SIM900	23
2.5.5 Năng lượng mặt trời	24
2.6 Kiến thức căn bản về ứng dụng web - di động và xây dựng Server . . .	24
2.6.1 Giao thức TCP	24
2.6.2 Căn bản về RESTful Web services	25
2.6.3 NodeJs	29

Mục lục	vi
2.6.4 Javascripts	31
2.6.5 Websocket - Socket.io	31
2.6.6 RethinkDB	33
2.6.7 Raspberry	35
2.6.8 Framework Ionic	36
3 Thiết kế và Hiện thực	37
3.1 Thiết kế hệ thống	37
3.1.1 Kiến trúc mô hình hệ thống	37
3.1.2 Các ràng buộc của hệ thống	37
3.1.3 Mô hình ứng dụng trình bày dữ liệu	38
3.2 Hiện thực Node cảm biến	38
3.2.1 Các node cảm biến thu thập dữ liệu	38
3.2.2 Ứng dụng theo dõi dữ liệu và đánh giá	38
3.3 Hệ thống Server lưu trữ dữ liệu và cung cấp API	39
3.3.1 Cấu trúc tổ chức tập tin	39
3.3.2 Thiết kế API	39
3.3.3 Xây dựng Web Server	40
3.4 Ứng dụng thiết bị di động	41
4 Kết quả thực nghiệm và Đánh giá	42
4.1 Mức độ tiêu hao năng	42
4.2 Tính ổn định của node cảm biến	42
4.3 Tính ổn định của Server	43
4.4 Đánh giá lập trình ứng dụng Mobile sử dụng APIs	43
5 Tổng kết và Hướng phát triển cho tương lai	44
5.1 Tổng kết	44
5.1.1 Những gì đạt được	44
5.1.2 Những khó khăn và hạn chế	44
5.2 Hướng phát triển tương lai	45
Tài liệu tham khảo	46
Chỉ mục	47

Danh sách hình vẽ

1.1	Mô hình IoT	2
2.1	Hình ảnh mô tả Internet of Things	5
2.2	Mô hình IoT không có Gateway	8
2.3	Mô hình IoT có Gateway	9
2.4	Kiến trúc mô hình IoT tham khảo	10
2.5	Ứng dụng IoT trong xây dựng	13
2.6	Ứng dụng IoT trong năng lượng	14
2.7	Ứng dụng IoT trong dân dụng	15
2.8	Ứng dụng IoT trong y tế	16
2.9	Trạm giám sát VOV Giao thông	20
2.10	Ứng dụng Web SmartBKTraffic	21
2.11	Trạm quan trắc môi trường tại Hà Nội	22
2.12	Cấu trúc của một đối tượng JSON	27
2.13	Cấu trúc mảng của JSON	27
2.14	Cấu trúc của một đối tượng JSON	28
2.15	Cấu trúc mô tả XML-JSON	29
2.16	Nền tảng Nodejs	30
2.17	Mô hình hoạt động Nodejs	30
2.18	Mô hình hoạt động Websocket	32
2.19	Biểu tượng của RethinkDB	33
2.20	Đoạn code mẫu RethinkDB	34
2.21	Thiết bị Raspberry Pi	35

Danh sách bảng

2.1 My caption	12
--------------------------	----

Chương 1

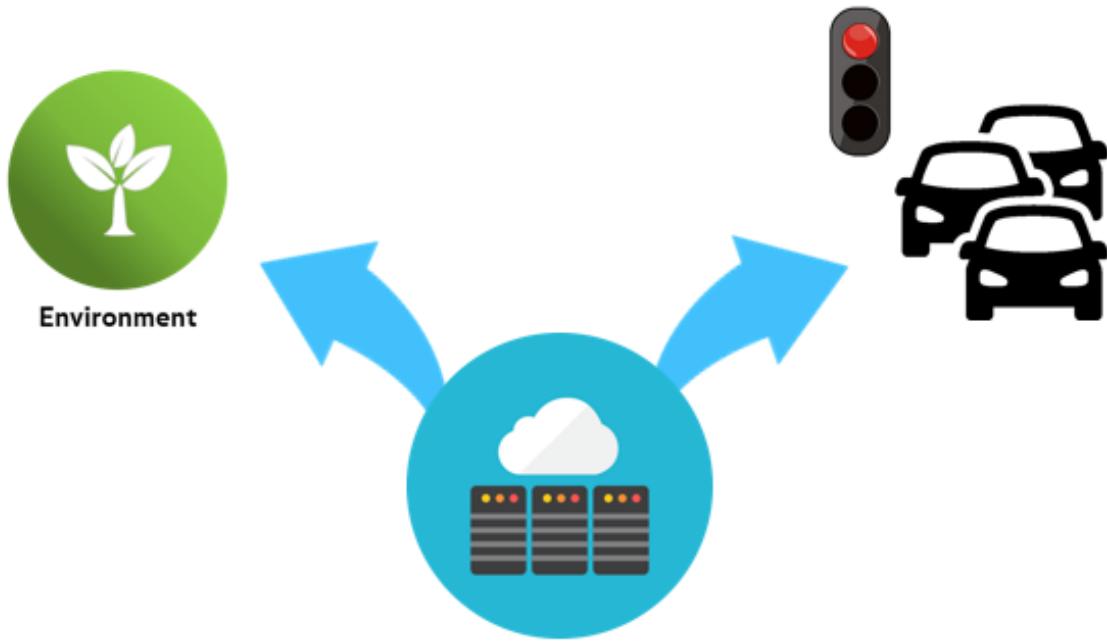
Giới thiệu

1.1 Ý tưởng và tính cấp thiết của đề tài

Ngày nay, bài toán giao thông vẫn đang là bài toán hóc búa vẫn chưa được giải quyết được ở Việt Nam và nhiều nước đang phát triển. Tình trạng kẹt xe, ùn tắc kéo dài gây ra sự chậm trễ trong công việc, hơn nữa còn gây gia tăng ô nhiễm môi trường, giảm chất lượng môi trường sống. Một trong những khó khăn làm bài toán giao thông khó giải quyết đó chính là không có đầy đủ dữ liệu cần thiết. Chúng ta không thể giải bài toán nếu không có đủ dữ kiện, cũng như giải quyết kẹt xe ta cần phải có dữ liệu về lưu lượng giao thông.

Vì yếu tố trên, dự án Smart Traffic được hình thành và phát triển theo hướng IoT với mục đích thu thập dữ liệu về các yếu tố môi trường (nồng độ CO, nhiệt độ, độ ẩm, nồng độ bụi, độ ồn...) và xác định mối tương quan giữa lưu lượng giao thông với môi trường xung quanh khu vực đó. Từ đó ta có thể có được 1 phần dữ liệu cần thiết về tình trạng các con đường theo thời gian thực, cũng như có được lịch sử và dùng dữ liệu ấy để phát triển dự đoán tình trạng giao thông tiếp theo.

Những dữ liệu đó có thể được phân tích và xử lý, đưa ra những kết luận về tình trạng lưu thông trên đoạn đường đó, về sự thay đổi môi trường tại 1 khu vực ở những



Hình 1.1 Mo ta tam hinh Mo hinh IoT

khoảng thời gian khác nhau. Và những dữ liệu này có thể ứng dụng cho bất động sản, môi trường và nghiên cứu. Do đó đề tài mang tính thiết thực và ứng dụng cao và được lựa chọn để làm Thực tập tốt nghiệp và có thể lên Luận văn tốt nghiệp.

1.2 Mục tiêu và nhiệm vụ của đề tài

1.2.1 Mục tiêu đề tài

- Tìm hiểu về khí thải xe máy và ô tô.
- Hiện thực mạch cảm biến để thu thập khí thải.
- Khảo sát và đo khí thải thực tế tại một số điểm thường xuyên xảy ra kẹt xe trên địa bàn thành phố.

- Tổng hợp các số liệu thu thập và các yếu tố môi trường khác (nhiệt độ, độ ẩm, ...) để đề xuất mô hình để hướng đến phát triển hệ thống cảnh báo kẹt xe dựa vào tình trạng ô nhiễm không khí.

1.2.2 Nhiệm vụ đề tài

Phân chia công việc tại đây!!!! Sony, Tùng, Cường làm những gì, vẽ giản đồ gantt.

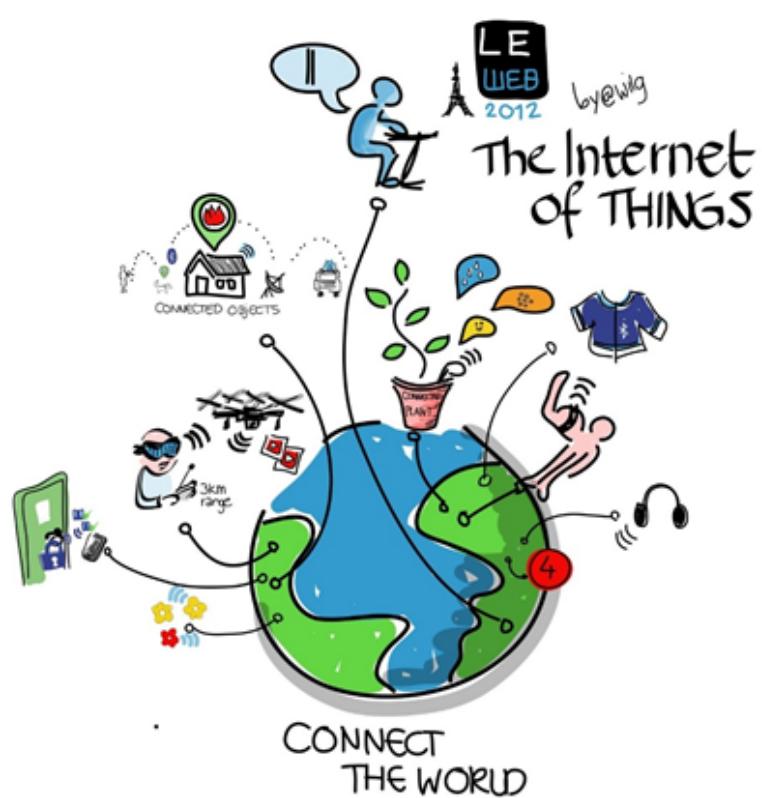
Chương 2

IoT và Các kiến thức cơ bản về IoT

2.1 Giới thiệu về Internet of Things và các ứng dụng giám sát

2.1.1 Internet of Things (IoT)

Với sự phát triển nhanh chóng của công nghệ máy tính hiện nay, những công nghệ cao đã đang và ngày càng được ứng dụng trong nhiều lĩnh vực của cuộc sống con người. IoT là mạng lưới của các đối tượng vật lý, các thiết bị, các phương tiện và các đối tượng này được nhúng với các thiết bị điện tử, cảm biến, phần mềm điều khiển, và nó có khả năng trao đổi dữ liệu và thao tác với nhau. IoT cho phép các đối tượng được lắng nghe và điều khiển từ xa trên cơ sở hạ tầng mạng hiện có, các hệ thống máy tính có thể thu thập dữ liệu và điều khiển các thiết bị đối tượng một cách hiệu quả, chính xác và lợi ích kinh tế nhất có thể.



Hình 2.1 Hình ảnh mô tả Internet of Things

Khả năng định danh độc nhất

Điểm quan trọng của IoT đó là các đối tượng phải có thể được nhận biết và định dạng. Nếu mọi đối tượng, kể cả con người, được "đánh dấu" để phân biệt bản thân đối tượng đó với những thứ xung quanh thì chúng ta có thể hoàn toàn quản lý được nó thông qua máy tính. Việc đánh dấu có thể được thực hiện thông qua nhiều công nghệ, chẳng hạn như RFID, NFC, mã vạch, mã QR, watermark kỹ thuật số... Việc kết nối thì có thể thực hiện qua Wi-Fi, mạng viễn thông băng rộng (3G, 4G), Bluetooth, ZigBee, hồng ngoại... Ngoài những kỹ thuật nói trên, nếu nhìn từ thế giới web, chúng ta có thể sử dụng các địa chỉ độc nhất để xác định từng vật, chẳng hạn như địa chỉ IP. Mỗi thiết bị sẽ có một IP riêng biệt không nhầm lẫn. Sự xuất hiện của IPv6 với không gian địa chỉ cực kỳ rộng lớn sẽ giúp mọi thứ có thể dễ dàng kết nối vào Internet cũng như kết nối với nhau.

Xu hướng và tính chất của The Internet of Things

Thông minh: Sự thông minh và tự động trong điều khiển thực chất không phải là một phần trong ý tưởng về IoT. Các máy móc có thể dễ dàng nhận biết và phản hồi lại môi trường xung quanh (ambient intelligence), chúng cũng có thể tự điều khiển bản thân (autonomous control) mà không cần đến kết nối mạng. Tuy nhiên, trong thời gian gần đây người ta bắt đầu nghiên cứu kết hợp hai khái niệm IoT và autonomous control lại với nhau. Tương lai của IoT có thể là một mạng lưới các thực thể thông minh có khả năng tự tổ chức và hoạt động riêng lẻ tùy theo tình huống, môi trường, đồng thời chúng cũng có thể liên lạc với nhau để trao đổi thông tin, dữ liệu.

Việc tích hợp trí thông minh vào IoT còn có thể giúp các thiết bị, máy móc, phần mềm thu thập và phân tích các dấu vết điện tử của con người khi chúng ta tương tác với những thứ thông minh, từ đó phát hiện ra các tri thức mới liên quan tới cuộc sống, môi trường, các mối tương tác xã hội cũng như hành vi con người.

Kiến trúc dựa trên sự kiện: Các thực thể, máy móc trong IoT sẽ phản hồi dựa theo các sự kiện diễn ra trong lúc chúng hoạt động theo thời gian thực. Một số nhà nghiên cứu từng nói rằng một mạng lưới các sensor chính là một thành phần đơn giản của IoT.

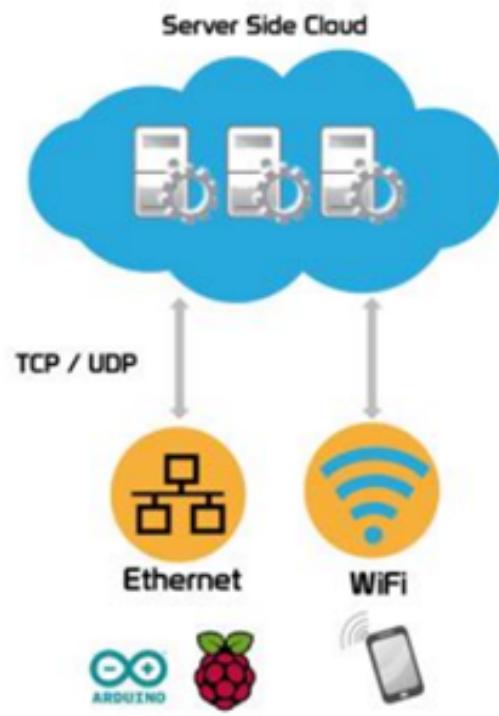
Là một hệ thống phức tạp: Trong một thế giới mở, IoT sẽ mang tính chất phức tạp bởi nó bao gồm một lượng lớn các đường liên kết giữa những thiết bị, máy móc, dịch vụ với nhau, ngoài ra còn bởi khả năng thêm vào các nhân tố mới.

Kích thước: Một mạng lưới IoT có thể chứa đến 50 đến 100 nghìn tỉ đối tượng được kết nối và mạng lưới này có thể theo dõi sự di chuyển của từng đối tượng. Một con người sống trong thành thị có thể bị bao bọc xung quanh bởi 1000 đến 5000 đối tượng có khả năng theo dõi.

Vấn đề không gian, thời gian: Trong IoT, vị trí địa lý chính xác của một vật nào đó là rất quan trọng. Hiện nay, Internet chủ yếu được sử dụng để quản lý thông tin được xử lý bởi con người. Do đó những thông tin như địa điểm, thời gian, không gian của đối tượng không mấy quan trọng bởi người xử lý thông tin có thể quyết định các thông tin này có cần thiết hay không, và nếu cần thì họ có thể bổ sung thêm. Trong khi đó, IoT về lý thuyết sẽ thu thập rất nhiều dữ liệu, trong đó có thể có dữ liệu thừa về địa điểm, và việc xử lý dữ liệu đó được xem như không hiệu quả. Ngoài ra, việc xử lý một khối lượng lớn dữ liệu trong thời gian ngắn đủ để đáp ứng cho hoạt động của các đối tượng cũng là một thách thức hiện nay.

Mô hình theo xu hướng IoT thực tế

Mô hình (Hình 2.2): Các thiết bị sẽ có thể kết nối với nhau bằng TCP/UDP và có thể kết nối trực tiếp đến máy chủ mà không cần thông qua thiết bị khác.



Hình 2.2 Mô hình IoT không có Gateway

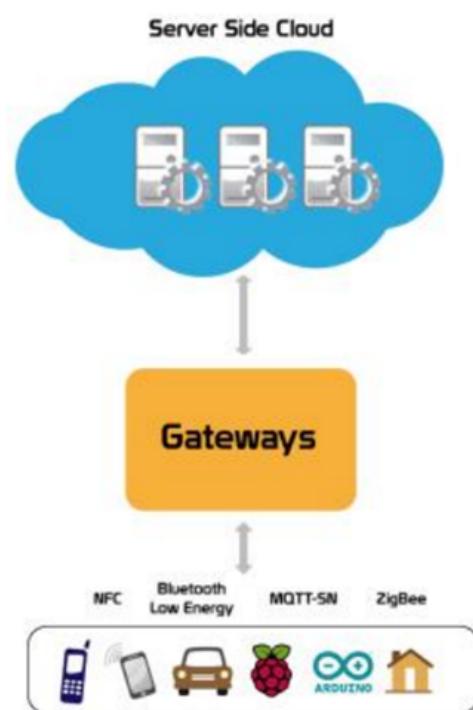
Mô hình (Hình 2.3): Các thiết bị sẽ được hỗ trợ nhiều chuẩn giao tiếp khác nhau (BLE, ZWave,...) nên cần cầu nối là Gateway để kết nối đến máy chủ.

Sơ đồ lớp Chung của cả hai mô hình trên:

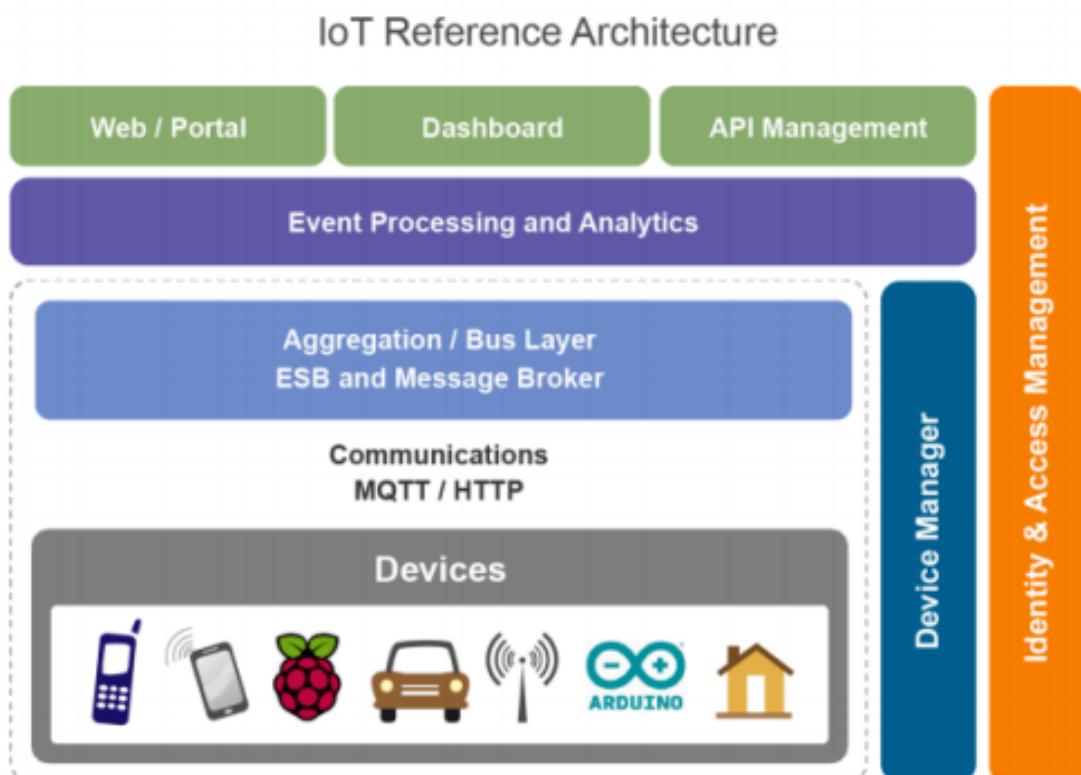
- Lớp Device

Lớp dưới cùng của kiến trúc là lớp thiết bị. Thiết bị có thể được các loại khác nhau, nhưng để có thể được xem là thiết bị IOT, nó phải có một số thông tin liên lạc hoặc gián tiếp hoặc trực tiếp với Internet.

Mỗi thiết bị thường cần một ID. ID có thể là: Bluetooth identifier, Wi-Fi MAC Address...



Hình 2.3 Mô hình IoT có Gateway



Hình 2.4 Kiến trúc mô hình IoT tham khảo

- Lớp Communications

Các lớp truyền thông hỗ trợ các kết nối của các thiết bị tới máy chủ. Có nhiều giao thức để giao tiếp giữa các thiết bị và máy chủ. Nổi bật nhất là HTTP hoặc MQTT.

HTTP rất thông dụng. Bởi vì nó là một giao thức dựa trên văn bản đơn giản, nhiều thiết bị nhỏ như bộ điều khiển 8-bit có thể hỗ trợ một phần các giao thức - ví dụ đủ dữ liệu để POST hoặc GET một nguồn tài nguyên. Các thiết bị lớn hơn 32-bit có thể sử dụng đầy đủ các thư viện HTTP client đúng cách để hiện thực toàn bộ giao thức.

MQTT được phát minh vào năm 1999 để giải quyết các vấn đề trong hệ thống nhúng và SCADA. Nó đã được thông qua một số lần lặp lại và phiên bản hiện tại (3.1.1) đang trải qua những tiêu chuẩn hóa trong OASIS MQTT kỹ thuật Committee8. MQTT là một hệ thống tin nhắn publish-subscribe dựa trên một mô hình broker. Các giao thức có chi phí rất nhỏ (ít nhất là 2 byte cho mỗi tin nhắn). MQTT được thiết kế để vận hành qua TCP.

- Lớp Aggregation/Bus

Một lớp quan trọng của kiến trúc là lớp mà tập hợp và là cầu nối, có khả năng tổng hợp và kết hợp các thông tin liên lạc từ các thiết bị khác nhau và đưa thông tin liên lạc đến một thiết bị cụ thể (có thể thông qua một gateway).

- Lớp Event Processing and Analytic:

Lớp này có các sự kiện từ lớp Bus và cung cấp khả năng xử lý và hành động theo những sự kiện này. Yêu cầu phải lưu trữ các dữ liệu vào một cơ sở dữ liệu nên buộc phải có một ứng dụng ở máy chủ.

- Lớp External Communications (Top)

Lớp này cung cấp một cách cho các thiết bị để chúng giao tiếp ra bên ngoài hệ

Bảng 2.1 My caption

MÔ HÌNH	ƯU ĐIỂM	NHƯỢC ĐIỂM
Mô hình cũ	Đơn giản hiện thực	Gateway cồng kềnh vì phải có bộ phận thu/phát hồng ngoại và các mạch để giao tiếp RF.
Mô hình 1	Không cần thông qua gateway, mọi thông tin thiết bị đều được lưu trên máy chủ.	Hạn chế về phương thức giao tiếp vì chỉ sử dụng TCP/UDP để giao tiếp. Không thể điều khiển thiết bị khi không truy cập được máy chủ.
Mô hình 2	Quản lý thiết bị thông qua gateway làm trung gian, nên các thiết bị có thể tương tác với nhau dễ dàng mà không cần đến máy chủ.	Chi phí cao hơn, hệ thống phức tạp hơn. Độ bảo mật cũng cần được quan tâm hơn.

thống một cách có định hướng và thân thiện với người dùng (Web, Dashboard, API Management)

2.1.2 Các hệ thống giám sát được phát triển dựa trên IoT

Khi kết nối một chuỗi khổng lồ các cảm biến (sensor) thu thập dữ liệu, thiết bị và máy móc với nhau, điều quan trọng cần nhận ra là thông tin sẽ được chuyển đổi thành hành động với một tốc độ mà chúng ta chưa từng thấy trước kia. Chúng ta đang tiến đến gần, nếu không phải là đã chạm được vào một thế giới của những khoảng thời gian phản ứng cực nhỏ, phản hồi tức thì với mọi điều kiện biến đổi, và mức độ điều khiển chưa từng có trong việc quản lý tài nguyên và tài sản. Điểm mấu chốt ở đây là đừng nghĩ hẹp. Internet of Things (IoT) không đơn thuần là mang đến sự tiết kiệm trong các mô hình công nghiệp hiện tại. Nó đảo lộn hoàn toàn những mô hình cũ, tạo ra những sản phẩm và dịch vụ mới. Không có một lĩnh vực nào mà trong đó IoT tạo ra ảnh hưởng đặc biệt lớn nhất; bởi IoT sẽ thay đổi hoàn toàn mọi lĩnh vực một cách

không thể tưởng tượng được, bao gồm nông nghiệp, năng lượng, an ninh, quản lý thảm họa, y tế, và đó chỉ là một vài lĩnh vực được nhắc đến.

Ứng dụng trong xây dựng



Hình 2.5 Ứng dụng IoT trong xây dựng

Ví dụ: Các công ty xây dựng đã bắt đầu trang bị các silo (hầm chứa đồ) và xe tải có các cảm biến theo dõi mức hàng tồn kho, như là lượng bê tông, và biến đổi nó thông qua platform trên nền điện toán đám mây để gia tăng tốc độ phân phối và đảm bảo một dòng lưu thông vật liệu ổn định. Các ông lớn trong ngành công nghiệp dầu mỏ đã bắt đầu thực thi các công nghệ mobile, cảm biến tới máy móc để dự phòng từ trước cho các tai nạn thông qua các phân tích nhanh chóng và hành động tức thời. Khi các cảm biến phát hiện ra một sự cố như rò rỉ hoặc thất thoát đường ống, công nghệ IoT cho phép các công nhân lập tức xác định vị trí của chúng.

Ứng dụng trong năng lượng



Hình 2.6 Ứng dụng IoT trong năng lượng

Một ví dụ khác của công nghệ IoT mới ứng dụng trong công nghiệp dầu mỏ, đó là giếng thông minh. Đây là một dạng giếng cài đặt các thiết bị điều khiển dòng chảy và cảm biến lỗ khoan, để có thể giám sát và điều khiển từ trên bề mặt mà không đe dọa an toàn của công nhân. Giếng thông minh có trang bị công nghệ địa chấn 4D, cho phép theo dõi sự rò rỉ khí ga, dòng chảy nước, thay đổi áp lực, và bất cứ thay đổi nào khác gây ra bởi những biến động địa chấn, giúp cho việc dự đoán và điều khiển các tác động địa chấn có thể gây ra những hỏng hóc nghiêm trọng. Nhưng như thế, chúng ta vẫn nghĩ quá hẹp. Hãy vượt ra khỏi lĩnh vực xây dựng hay năng lượng. Chúng ta có các cảm biến có thể đo lực, tải, moment, và áp lực; các cảm biến có thể ngửi thấy mùi khí ga hay hóa chất; những cảm biến có thể nghe thấy rung động và phân biệt giữa các âm hưởng khác nhau; những cảm biến có thể đo nhiệt độ, phát hiện chuyển động, vận tốc và chuyển vị; xác định vị trí, sự có mặt, và khoảng cách. Nói cách khác, chúng ta có khả năng thu thập những hiểu biết gần như không giới hạn, trong thời gian thực.



Hình 2.7 Ứng dụng IoT trong dân dụng

Ứng dụng trong dân dụng

Làm thế nào chúng ta có thể tận dụng thông tin thời gian thực từ rất nhiều sensor? Hãy nhìn vào ngôi nhà của chúng ta. Những phần nào trong đó có thể thông minh hóa? Ví dụ đơn giản. Tôi từng quan sát 1 hệ thống video conference cho phép người chủ nói chuyện với chú cún của mình, gọi nó đến, cho nó ăn từ xa thông qua một thiết bị thông minh. Hãy nghĩ lớn hơn nữa. Một ngôi nhà biết khi nào bạn về nhà bởi nó kết nối với một cảm biến trên xe hay smartphone của bạn. Một ngôi nhà kết nối các cảm biến báo khói, hệ thống an ninh, và thiết bị giải trí tới điện thoại của bạn. Một ngôi nhà với các cảm biến được gắn vào đường ống để có thể phát hiện ra rò rỉ trước cả khi điều đó thực sự diễn ra.

Ứng dụng trong y tế



Hình 2.8 Ứng dụng IoT trong y tế

Công nghệ thiết bị đeo cũng sẽ biến đổi hoàn toàn lĩnh vực chăm sóc sức khỏe theo những cách phi thường. Chúng ta đều biết rằng đồng hồ Apple Watch sẽ tích hợp một

cảm biến theo dõi nhịp tim và cung cấp cho chủ của nó những ứng dụng tạo điều kiện và khuyến khích một cách sống lành mạnh. Chúng ta đã có các cảm biến gắn trong giày để theo dõi việc chạy xa đến đâu và bao nhiêu calo đã được đốt. Còn gì tiếp theo? Sẽ có một quy trình tối ưu chăm sóc sức khỏe, theo đó có những cảm biến có thể phát hiện vi khuẩn trong thiết bị, và thiết bị diệt khuẩn phát hiện virus có thể di chuyển từ bệnh nhân.

2.2 Giới thiệu về các công cụ hỗ trợ phát triển ứng dụng IoT

2.3 Những yếu tố ảnh hưởng môi trường từ khí thải phương tiện giao thông

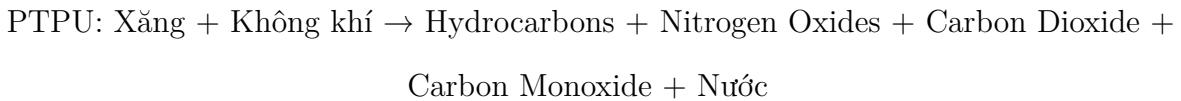
Để hình dung được những yếu tố ảnh hưởng môi trường từ khí thải xe máy và ô tô thì chúng ta cần biết được quá trình hoạt động của xe máy và ô tô, từ đó chúng ta sẽ biết được những loại khí thải nào mà xe máy và ô tô sẽ thải ra môi trường. Qua quá trình tìm hiểu và đọc thông tin tài liệu trên mạng khí thải của xe máy và ô tô tuy thuộc chủ yếu vào chất lượng đốt cháy hỗn hợp xăng và không khí bên trong buồng đốt (combustion chamber) của động cơ, cũng như nồng độ các chất ô nhiễm trong khí xả phụ thuộc vào đặc điểm động cơ cũng như các thông số điều chỉnh, vận hành.

Động cơ mới và được điều chỉnh đúng cho phản ứng cháy hoàn chỉnh (complete combustion) hay phản ứng cháy thừa oxy:



Các phó sản (by-product) chủ yếu của phản ứng này là nước (H_2O) và carbon dioxide (CO_2), do đó ống thoát khí cháy (tail pipe) của một động cơ tốt thường có nước nhieu

ra, dễ nhận thấy khi động cơ đang trong quá trình làm nóng máy (warm up). Động cơ cũ hoặc không được điều chỉnh đúng cho phản ứng cháy không hoàn chỉnh (incomplete combustion) hay thiếu oxy:



Phản ứng này tạo thêm những phó sản như carbon monoxide (CO) và nitrogen oxides (NOx). Vậy chúng ta có các loại sản phẩm như sau:

- CO được sinh ra khi lượng oxy đưa vào buồng đốt không đủ.
- HC được sinh ra trong quá trình đốt cháy không hoàn toàn, cũng như CO.
- NOx được sinh ra do nitơ và ôxy trong hỗn hợp không khí-nhiên liệu, khi nhiệt độ của buồng đốt tăng cao trên 1800oC.

Nhiệt độ của buồng đốt càng cao, lượng NOx sản ra càng nhiều. Theo lý thuyết, khi đốt cháy xăng thì chỉ sinh ra CO₂ (cácbon điôxit) và H₂O (hơi nước). Tuy nhiên, không phải toàn bộ xăng đều tham gia phản ứng như lí thuyết, do ảnh hưởng của các yếu tố như tỷ lệ hỗn hợp không khí-nhiên liệu, nitơ trong không khí, nhiệt độ cháy, thời gian cháy... Đó là nguyên nhân sinh ra các khí độc hại như CO, HC hoặc NOx. Trên cơ sở thực tế, động cơ đốt trong nó chỉ sử dụng được khoảng 30-45% nhiệt lượng để sinh công, phần còn lại bị hao hụt đi mất, do đó nó mang theo nhiệt lượng thải ra ngoài. Lượng nhiệt này thực tế đã tác động đến môi trường, làm cho nhiệt độ xung quanh những khu vực có xe máy và ô tô tăng lên. Việc đi lại trên những đoạn đường đông xe máy và ô tô thường gây cho chúng ta cảm giác nóng bức, và khó thở. Đó chính là những yếu tố khí thải đã trình bày như trên của xe máy và ô tô đã tác động đến môi trường. Một phần nhân tố cũng đáng chú ý trong quá trình hoạt động xe máy và ô tô hoạt động đó là bụi, vì đầu vào của động cơ là xăng và không khí, mà trong xăng

hiện nay thường có cặn và lượng khói bụi không khí đầu vào cũng có bụi mặc dù có bộ lọc khí nhưng không thể tránh được trong quá trình sử dụng lâu dài. Một số loại phương tiện giao thông sử dụng thời gian dài, không đi bảo trì sẽ thải ra một nồng độ ô nhiễm rất lớn. Từ những cơ sở lý thuyết trên cho chúng ta biết được những yếu tố ảnh hưởng môi trường từ khí thải xe máy và ô tô bao gồm:

- Khí CO₂, CO, HC, NOx.
- Khói bụi.
- Yếu tố về nhiệt.

2.4 Các hệ thống quan trắc hiện hữu

VoV giao thông: sử dụng hệ thống camera an ninh kết hợp với lượng phóng viên thường trực khắp thành phố để giám sát trực tiếp và cảnh báo tình trạng lưu thông trên các đoạn đường. Hệ thống có tính hiệu quả rất cao vì sử dụng CCTV giám sát tình trạng giao thông theo thời gian thực nên có góc nhìn thực tế nhất và không phục thuộc, có thể hoạt động độc lập. Tuy nhiên hệ thống vẫn có nhiều hạn chế như:

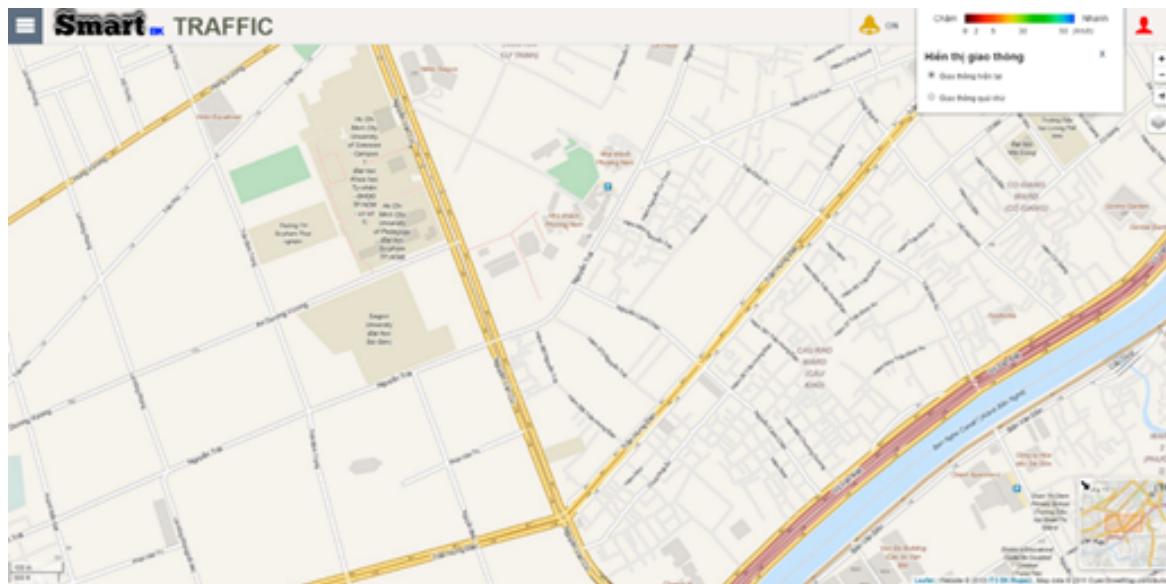
- Vẫn chủ yếu hoạt động một cách thủ công, điều tiết và đưa ra kết quả bởi con người, chưa có áp dụng thuật toán xử lý máy tính vào công việc nhiều.
- Tốn nhiều công sức và nhân lực, cần có nhiều phóng viên trực thuộc các đoạn đường cũng như người quản lý theo dõi hệ thống camera giao thông.
- Chi phí duy trì và lắp đặt còn khá cao, và cần tốn chi phí bảo trì đắt đỏ và trình độ chuyên môn nhân viên điều hành cao, chi phí một hệ thống có thể lên tới hàng trăm tỉ. Cụ thể như hệ thống trị giá 270 tỉ VNĐ bao gồm 19 trạm giám sát camera được triển khai ở Bà Rịa - Vũng Tàu.



Hình 2.9 Trạm giám sát VOV Giao thông

Bktraffic <http://traffic.hcmut.edu.vn> kết hợp hệ thống xe bus cùng với định vị GPS và gửi dữ liệu qua hạ tầng mạng 3G để theo dõi vị trí xe và tính toán tốc độ lưu thông trên đoạn đường xe đang lưu thông. Dự án này có hiệu quả cao vì hệ thống xe bus dày đặc và có thời gian hoạt động rộng. Tuy nhiên vẫn có những hạn chế như:

- Vì chỉ giám sát trên xe bus nhưng nhiều trường hợp không hiệu quả khi áp dụng cho giao thông tại Việt Nam. Bởi vì đa số phương tiện giao thông ở Việt Nam là xe 2 bánh, do đó hệ thống bus di chuyển không phản ánh được tình trạng lưu thông chung trên tuyến đường đó.
- Nhiều tuyến đường trên địa bàn thành phố mà xe bus vẫn chưa hoạt động, do vậy vẫn chưa có đủ dữ liệu cần thiết để.
- Mang tính phụ thuộc vào xe bus, vậy nên có thể có những trường hợp ảnh hưởng bởi hệ thống xe bus mà hệ thống sẽ không hoạt động hiệu quả được.



Hình 2.10 Ứng dụng Web SmartBKTraffic

Hệ thống quan trắc môi trường ở Hà Nội: Thủ đô Hà Nội vừa lắp đặt hệ thống 80 trạm quan trắc. Hệ thống này có tính tương đồng với đề tài đang thực hiện.

Tuy nhiên mục đích chính là để theo dõi môi trường sau sự cố bụi Thủy ngân và chưa thấy hướng ứng dụng vào hệ thống giao thông. Hơn nữa, các trạm này chiếm diện tích lắp đặt nên hạn chế khả năng triển khai số lượng lớn trên diện rộng.



Hình 2.11 Trạm quan trắc môi trường tại Hà Nội

2.5 Các thiết bị phần cứng

2.5.1 Thiết bị thu thập dữ liệu

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

2.5.2 Các loại sensor

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

2.5.3 Mạch vi xử lý Arduino

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

2.5.4 Module SIM900

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

2.5.5 Năng lượng mặt trời

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

2.6 Kiến thức căn bản về ứng dụng web - di động và xây dựng Server

2.6.1 Giao thức TCP

TCP (Transmission Control Protocol - "Giao thức điều khiển truyền vận") là một trong các giao thức cốt lõi của bộ giao thức TCP/IP. Sử dụng TCP, các ứng dụng trên các máy chủ được nối mạng có thể tạo các "kết nối" với nhau, mà qua đó chúng có thể trao đổi dữ liệu hoặc các gói tin. Giao thức này đảm bảo chuyển giao dữ liệu tới nơi nhận một cách đáng tin cậy và đúng thứ tự. TCP còn phân biệt giữa dữ liệu của nhiều ứng dụng (chẳng hạn, dịch vụ Web và dịch vụ thư điện tử) đồng thời chạy trên cùng một máy chủ.

TCP hỗ trợ nhiều giao thức ứng dụng phổ biến nhất trên Internet và các ứng dụng kết quả, trong đó có WWW, thư điện tử và Secure Shell.

Trong bộ giao thức TCP/IP, TCP là tầng trung gian giữa giao thức IP bên dưới và một ứng dụng bên trên. Các ứng dụng thường cần các kết nối đáng tin cậy kiểu đường ống để liên lạc với nhau, trong khi đó, giao thức IP không cung cấp những dòng kiểu

đó, mà chỉ cung cấp dịch vụ chuyển gói tin không đáng tin cậy. TCP làm nhiệm vụ của tầng giao vận trong mô hình OSI đơn giản của các mạng máy tính.

Các ứng dụng gửi các dòng gồm các byte 8-bit tới TCP để chuyển qua mạng. TCP phân chia dòng byte này thành các đoạn (segment) có kích thước thích hợp (thường được quyết định dựa theo kích thước của đơn vị truyền dẫn tối đa (MTU) của tầng liên kết dữ liệu của mạng mà máy tính đang nằm trong đó). Sau đó, TCP chuyển các gói tin thu được tới giao thức IP để gửi nó qua một liên mạng tới mô đun TCP tại máy tính đích. TCP kiểm tra để đảm bảo không có gói tin nào bị thất lạc bằng cách gán cho mỗi gói tin một "số thứ tự" (sequence number). Số thứ tự này còn được sử dụng để đảm bảo dữ liệu được trao cho ứng dụng đích theo đúng thứ tự. Mô đun TCP tại đầu kia gửi lại "tin báo nhận" (acknowledgement) cho các gói tin đã nhận được thành công; một "đồng hồ" (timer) tại nơi gửi sẽ báo time-out nếu không nhận được tin báo nhận trong khoảng thời gian bằng một round-trip time (RTT), và dữ liệu (được coi là bị thất lạc) sẽ được gửi lại. TCP sử dụng checksum (giá trị kiểm tra) để xem có byte nào bị hỏng trong quá trình truyền hay không; giá trị này được tính toán cho mỗi khối dữ liệu tại nơi gửi trước khi nó được gửi, và được kiểm tra tại nơi nhận.

2.6.2 Căn bản về RESTful Web services

Tổng quan về REST – Representational State Transfer

REST là một kiểu kiến trúc dựa trên các tiêu chuẩn web và giao thức HTTP. REST được mô tả đầu tiên bởi Roy Fielding trong luận văn tiến sĩ của ông vào năm 2000.

Trong một kiến trúc REST thì tất cả mọi thứ là một nguồn tài nguyên. Một nguồn tài nguyên được truy cập thông qua một giao diện phổ biến dựa trên các phương pháp tiêu chuẩn HTTP.

Một kiến trúc REST thường có một REST server cung cấp truy cập đến các nguồn tài nguyên và một Rest client truy cập và trình sửa các tài nguyên đó.

REST cho phép các nguồn tài nguyên biểu diễn dưới nhiều kiểu khác nhau như Text, XML, JSON ... Rest client có thể yêu cầu một kiểu biểu diễn cụ thể thông qua giao thức HTTP.

Các phương thức HTTP

Các phương thức PUT, GET, POST và DELETE thường được sử dụng trong kiến trúc REST

- **GET** định nghĩa một truy cập đọc nguồn tài nguyên mà không có tác dụng phụ. Nguồn tài nguyên không bao giờ thay đổi thông qua một yêu cầu GET.
- **PUT** dùng để tạo một tài nguyên mới.
- **DELETE** loại bỏ các nguồn tài nguyên.
- **POST** cập nhật một nguồn tài nguyên hiện có hoặc tạo ra một nguồn tài nguyên mới.

RESTful webservice

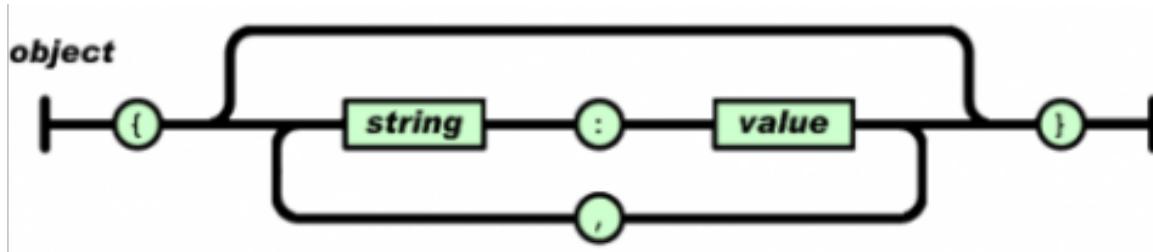
Một RESTful webservice được dựa trên các phương thức HTTP và các khái niệm về REST. Một RESTful webservice thường xác định URI cơ sở cho các dịch vụ, hỗ trợ các kiểu MIME (XML, Text, JSON,...) và tập các hoạt động (POST, GET, PUT, DELETE) đều được hỗ trợ.

JSON (JavaScript Object Notation)

Là một định dạng hoán vị dữ liệu nhanh. Chúng dễ dàng cho chúng ta đọc và viết. Dễ dàng cho thiết bị phân tích và phát sinh. Chúng là cơ sở dựa trên tập hợp của Ngôn Ngữ Lập Trình JavaScript, tiêu chuẩn ECMA-262 phiên bản 3 – tháng 12 năm 1999. JSON là một định dạng kiểu text mà hoàn toàn độc lập với các ngôn ngữ hoàn chỉnh, thuộc họ hàng với các ngôn ngữ họ hàng C, gồm có C, C++, C, Java, JavaScript, Perl, Python, và nhiều ngôn ngữ khác. Những đặc tính đó đã tạo nên JSON một ngôn ngữ hoán vị dữ liệu lý tưởng.

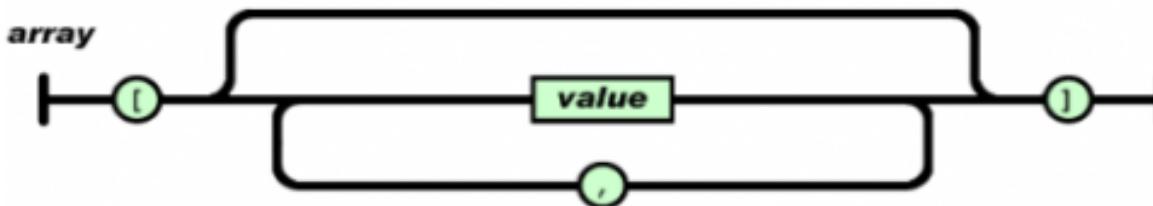
JSON được xây dựng trên 2 cấu trúc:

- Là tập hợp của các cặp tên và giá trị name – value. Trong những ngôn ngữ khác nhau, đây được nhận thấy như là một đối tượng (object), sự ghi (record), cấu trúc (struct), từ điển (dictionary), bảng băm (hash table), danh sách khóa (keyed list), hay mảng liên hợp.
- Là một tập hợp các giá trị đã được sắp xếp. Trong hầu hết các ngôn ngữ, nó được nhận thấy như là một mảng, vector, list hay là một dãy sequence.



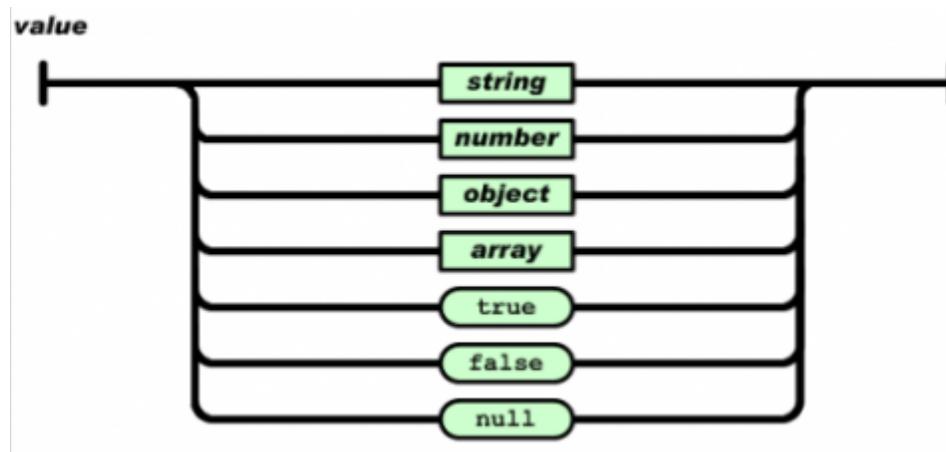
Hình 2.12 Cấu trúc của một đối tượng JSON

Một đối tượng là một tập hợp của các cặp tên/giá trị (name/value). Một đối tượng bắt đầu bởi dấu “{” và kết thúc với dấu “}”. Từng tên được sau bởi dấu “:” và các cặp tên/giá trị được tách ra bởi dấu “,”.



Hình 2.13 Cấu trúc mảng của JSON

Một mảng là một tập hợp các giá trị đã được xắp xếp, một mảng bắt đầu bởi dấu “[” và kết thúc với dấu ”]”. Các giá trị được cách nhau bởi dấu “,”.



Hình 2.14 Cấu trúc của một đối tượng JSON

Một giá trị có thể là chuỗi, số, đúng sai, hoặc là đối tượng hoặc mảng. Những cấu trúc này có thể được lồng vào nhau.

JSON và Webservice

Việc trao đổi thông tin, truyền tải dữ liệu giữa Webservice và ứng dụng người ta thường dùng XML hay JSON vì sự tiện nghi và cấu trúc dữ liệu nhỏ gọn của nó. JSON là một ngôn ngữ mới với tuổi đời con trẻ nhưng vẫn được ứng dụng nhiều bởi những điểm lợi của nó:

- “Con người” có thể đọc được.
- Cú pháp gần với JavaScript nên rất dễ để sử dụng.
- Dữ liệu truyền tải ngắn gọn so với những định dạng dữ liệu như: XML, HTML...
- Việc phân tích (parse) dữ liệu từ dạng chuỗi (nhận từ server) sang dữ liệu có thể sử dụng được (thành Object, Number, Array) dễ dàng.

The screenshot shows two side-by-side code editors. The left editor is titled "personal.xml" and contains the following XML code:

```

<?xml version="1.0" encoding="UTF-8"?>
<personnel>
    <person id="Big.Boss">
        <name>
            <family>Boss</family>
            <given>Big</given>
        </name>
        <email>chief@oxygencode.com</email>
        <link subordinates="one.worker"/>
    </person>
    <person id="one.worker">
        <name>
            <family>Worker</family>
            <given>One</given>
        </name>
        <email>one@oxygencode.com</email>
        <link manager="Big.Boss"/>
    </person>
    <person id="two.worker">
        <name>
            <family>Worker</family>
            <given>Two</given>
        </name>
        <email>two@oxygencode.com</email>
        <link manager="Big.Boss"/>
    </person>
    <person id="three.worker">
        <name>
    
```

The right editor is titled "personal.json" and contains the following JSON code:

```

{
    "personnel": [
        {
            "id": "Big.Boss",
            "name": {
                "family": "Boss",
                "given": "Big"
            },
            "email": "chief@oxygencode.com",
            "link": {"subordinates": "one.worker"}
        },
        {
            "id": "one.worker",
            "name": {
                "family": "Worker",
                "given": "One"
            },
            "email": "one@oxygencode.com",
            "link": {"manager": "Big.Boss"}
        },
        {
            "id": "two.worker",
            "name": {
                "family": "Worker",
                "given": "Two"
            },
            "email": "two@oxygencode.com",
            "link": {"manager": "Big.Boss"}
        }
    ]
}

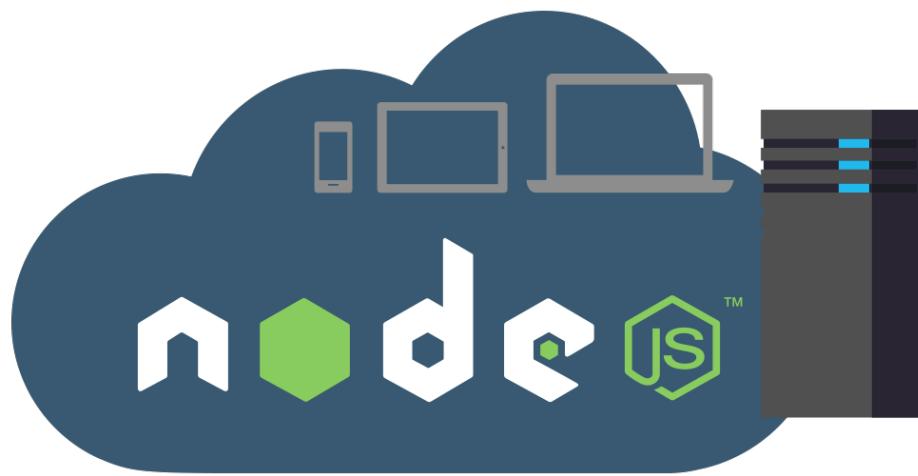
```

Hình 2.15 Cấu trúc mô tả XML-JSON

Nhằm mục đích giúp thời gian tương tác giữa người sử dụng với ứng dụng tiến gần đến thời gian thực, thì tốc độ được đặt lên hàng đầu, việc tăng tốc bằng cách “giảm tải” cũng là một giải pháp, chính vì vậy JSON được ưu ái hơn vì cú pháp ngắn gọn, đơn giản, tiết kiệm hơn XML.

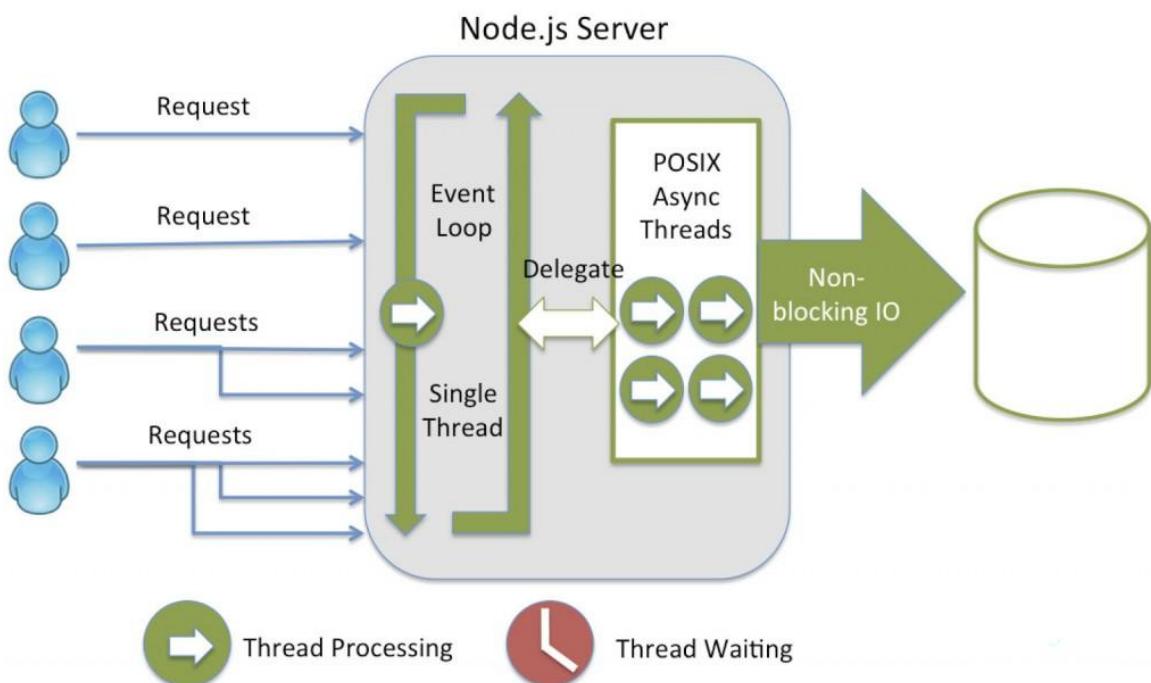
2.6.3 NodeJS

NodeJS là một nền tảng Server side được xây dựng dựa trên Javascript Engine (V8 Engine). Node.js được phát triển bởi Ryan Dahl năm 2009. Định nghĩa Nodejs bởi tài liệu chính thức : Node.js là một nền tảng dựa vào Chrome Javascript runtime để xây dựng các ứng dụng nhanh, có độ lớn.



Hình 2.16 Nền tảng Nodejs

Node.js sử dụng các phần phát sinh các sự kiện (event-driven), mô hình non-blocking I/O để tạo ra các ứng dụng nhẹ và hiệu quả cho các ứng dụng về dữ liệu thời gian thực chạy trên các thiết bị phân tán.



Hình 2.17 Mô hình hoạt động Nodejs

2.6.4 Javascripts

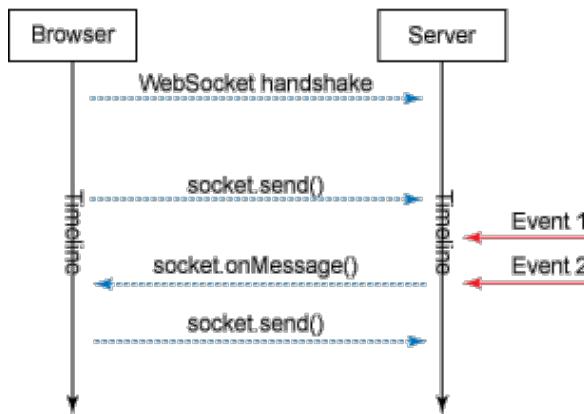
JavaScript, theo phiên bản hiện hành, là một ngôn ngữ lập trình kịch bản dựa trên đối tượng được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web, nhưng cũng được dùng để tạo khả năng viết script sử dụng các đối tượng nằm sẵn trong các ứng dụng. Nó vốn được phát triển bởi Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java. .js là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript.

Những ứng dụng to lớn của Javascript khiến người ta không thể quên nó được. Hiện nay có rất nhiều libraries, Framework được viết như:

- AngularJS: Một thư viện dùng để xây dựng ứng dụng Single Page
- NodeJS: Một thư viện được phát triển phía Server dùng để xây dựng ứng dụng realtime
- Sencha Touch: Một Framework dùng để xây dựng ứng dụng Mobile
- ExtJS: Một Framework dùng xây dựng ứng dụng quản lý (Web Applications)
- jQuery: Một thư viện rất mạnh về hiệu ứng

2.6.5 Websocket - Socket.io

WebSoket là công nghệ hỗ trợ giao tiếp hai chiều giữa client và server bằng cách sử dụng một TCP socket để tạo một kết nối hiệu quả và ít tốn kém. Mặc dù được thiết kế để chuyên sử dụng cho các ứng dụng web, lập trình viên vẫn có thể đưa chúng vào bất kì loại ứng dụng nào.



Hình 2.18 Mô hình hoạt động Websocket

Kết nối được duy trì có thể viết và nhận dữ liệu bằng JavaScript như khi bạn đang sử dụng một TCP socket đơn thuần.

- HTTP: $871 \times 10,000 = 8,710,000$ bytes = 69,680,000 bits per second (66 Mbps) -
WebSocket: $2 \times 10,000 = 20,000$ bytes = 160,000 bits per second (0.153 Kbps)

Socket.IO

Socket.IO là một thư viện javascript có mục đích tạo ra các ứng dụng realtime trên trình duyệt cũng như thiết bị di động. Việc sử dụng thư viện này cũng rất đơn giản và giống nhau ở cả server lẫn client.

Tại sao Socket.IO ra đời?

- Javascript là ngôn ngữ lập trình hướng sự kiện, mà trong lập trình thời gian thực, cách tiếp cận bằng lập trình sự kiện là cách tiếp cận khôn ngoan nhất.
- Node.js chạy non-blocking việc hệ thống không phải tạm ngừng để xử lý xong một request sẽ giúp cho server trả lời client gần như ngay tức thì
- Lập trình socket yêu cầu bạn phải xây dựng được mô hình lắng nghe – trả lời từ cả 2 bên. Nói khác đi, vai trò của client và server phải tương đương nhau, mà client thì chạy bằng javascript, nên nếu server cũng chạy bằng javascript nữa, thì việc lập trình sẽ dễ dàng và thân thiện hơn.

Socket.IO hoạt động dựa trên các events tương tự như Websocket:

- **connect()**: kết nối với server socket
- **on(event_name, listener)**: đăng ký lắng nghe sự kiện từ server trả về
- **emit(event_name, data)**: gửi một sự kiện lên server
- **off(event_name)**: ngừng lắng nghe một sự kiện nào đó

2.6.6 RethinkDB

RethinkDB là gì?

RethinkDB là một mã nguồn mở, dựa theo cấu trúc cơ sở dữ liệu NoSQL, tổ chức dữ liệu theo kiểu document-oriented database - một thiết kế riêng biệt cho việc lưu trữ document. Các cài đặt có thể là giả lập tương tác trên relational database, object database hay key-value store. Cơ sở dữ liệu được lưu trữ dưới dạng JSON với các giao diện động, và được thiết kế giúp hoạt động dễ dàng trong việc cập nhật và truy xuất dữ liệu theo thời gian thực với các ứng dụng.



Hình 2.19 Biểu tượng của RethinkDB

RethinkDB sử dụng ngôn ngữ truy vấn ReQL được phát triển theo hướng đối tượng. Và ngôn ngữ này hỗ trợ cho các thao tác nhập bảng, quản lý nhóm, tích hợp và các hàm chức năng truy vấn cũng như event theo thời gian thực.

The screenshot shows a code editor window with the following code:

```
r.table('game').orderBy('score').limit(3).changes()
```

Below the code, there are two sections:

- STREAMING RETHINKDB RESULTS...**: This section displays a stream of JSON objects representing player scores. The output is as follows:


```
{"player": "matt", "score": 129}
{"player": "tim", "score": 115}
{"player": "tim", "score": 146}
{"player": "yoshiko", "score": 154}
{"player": "tim", "score": 140}
{"player": "pierre", "score": 67}
{"player": "anthony", "score": 112}
{"player": "connor", "score": 53}
{"player": "marshall", "score": 126}
```
- TOP PLAYER SCORES**: This section lists the top three players and their scores:
 1. yoshiko: 154 points
 2. tim: 146 points
 3. tim: 140 points

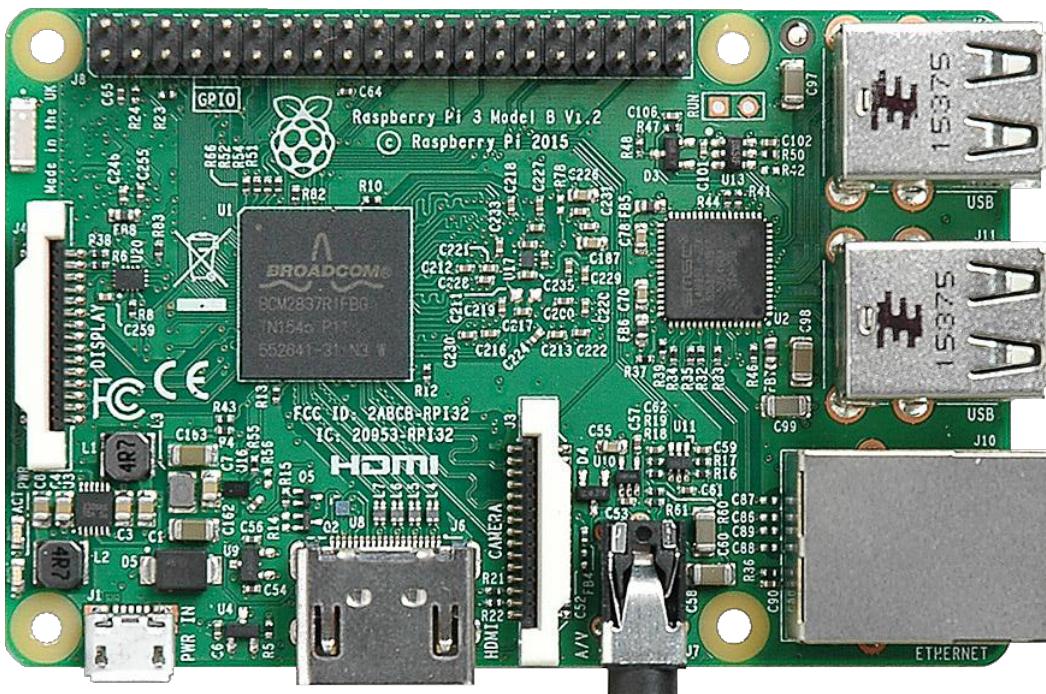
Hình 2.20 Đoạn code mẫu RethinkDB

Như hình 2.20 là ví dụ trực quan được demo trực tiếp tại trang chủ <https://www.rethinkdb.com>, có thể nhận thấy được cách thức khai báo và sử dụng RethinkDB:

- **r** : là đối tượng RethinkDB chúng ta khai báo trong ứng dụng.
- **table()** : là bảng giá trị đang truy vấn.
- **orderBy()** : hàm sắp xếp dữ liệu theo trường giá trị được khai báo.
- **limit()** : chọn ra những giá trị trong phạm vi được khai báo trong hàm.
- **changes()** : đây chính là đặc điểm khiến RethinkDB khác biệt so với các hệ quản lý dữ liệu khác. RethinkDB sẽ gọi hàm thực thi khi vùng giá trị truy vấn có thay đổi giá trị theo thời gian thật. không hoạt động theo cơ chế polling thường thấy ở các kiểu truy vấn dữ liệu khác.

2.6.7 Raspberry

Raspbian là hệ điều hành mở và miễn phí được phát triển tối ưu hóa dựa trên nền Debian dành riêng cho thiết bị phần cứng Raspberry Pi. Hệ điều hành này bao gồm nhóm các chức năng và ứng dụng giúp Raspberry hoạt động. Hơn nữa, Raspbian cung cấp nhiều hơn là hệ điều hành thuần: hỗ trợ tới hơn 35,000 gói mở rộng và dễ dàng cài đặt trên Raspberry Pi. Raspbian ổn định và hiệu suất cao được hoàn thiện vào tháng 6 năm 2012. Tuy nhiên Raspbian vẫn đang được phát triển tới ngày nay và ngày càng cải thiện hiệu suất cũng như hỗ trợ các gói chức năng nhiều nhất có thể.



Hình 2.21 Thiết bị Raspberry Pi

Vì sao lựa chọn Raspberry:

- Là một máy tính độc lập chạy trên nền tảng Linux, điều này làm cho việc chạy các ứng dụng Node.js hay RethinkDB hoạt động được trên Raspberry Pi.
- Hiệu suất đáp ứng nhu cầu làm server của đề tài yêu cầu với mức độ tiêu thụ năng lượng cực kì ít.

2.6.8 Framework Ionic

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Chương 3

Thiết kế và Hiện thực

3.1 Thiết kế hệ thống

3.1.1 Kiến trúc mô hình hệ thống

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

3.1.2 Các ràng buộc của hệ thống

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

3.1.3 Mô hình ứng dụng trình bày dữ liệu

 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

3.2 Hiện thực Node cảm biến

3.2.1 Các node cảm biến thu thập dữ liệu

 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

3.2.2 Ứng dụng theo dõi dữ liệu và đánh giá

 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

3.3 Hệ thống Server lưu trữ dữ liệu và cung cấp API

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

3.3.1 Cấu trúc tổ chức tập tin

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

3.3.2 Thiết kế API

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

```
1  Name.prototype = {  
2    methodName: function(params){  
3      var doubleQuoteString = "some text";  
4      var singleQuoteString = 'some more text';  
5      // this is a comment  
6      if(this.confirmed != null && typeof(this.confirmed) == Boolean &&  
7          this.confirmed == true){  
8        document.createElement('h3');  
9        $('#system').append("This looks great");  
10       return false;  
11     } else {  
12       throw new Error;  
13     }  
14   }
```

Listing 3.1 My Javascript Example

3.3.3 Xây dựng Web Server

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

3.4 Ứng dụng thiết bị di động

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Chương 4

Kết quả thực nghiệm và Đánh giá

4.1 Mức độ tiêu hao năng

nói về đo đạc v.v chõ này nên bỏ bên hiện thực hay kết quả ?

 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

4.2 Tính ổn định của node cảm biến

này nói về mấy tháng đo, gió mưa đồ

 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

4.3 Tính ổn định của Server

Nói về thời gian đáp ứng truyền nhận dữ liệu

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

4.4 Đánh giá lập trình ứng dụng Mobile sử dụng APIs

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Chương 5

Tổng kết và Hướng phát triển cho tương lai

5.1 Tổng kết

5.1.1 Những gì đạt được

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

5.1.2 Những khó khăn và hạn chế

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure

dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum And now I begin my third chapter here . . .

5.2 Hướng phát triển tương lai

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Tài liệu tham khảo

- [1] Ancey, C., Coussot, P., and Evesque, P. (1996). Examination of the possibility of a fluid-mechanics treatment of dense granular flows. *Mechanics of Cohesive-frictional Materials*, 1(4):385–403.
- [2] Read, C. J. (1985). A solution to the invariant subspace problem on the space l_1 . *Bull. London Math. Soc.*, 17:305–317.

Chỉ mục

LaTeX class file, 1