**BỘ CÔNG THƯƠNG**

**Chương trình quốc gia phát triển công nghệ cao đến năm 2020**

**Chương trình phát triển một số ngành công nghiệp công nghệ cao**

**BÁO CÁO CHUYÊN ĐỀ SỐ X.Y.Z:**

**“Nghiên cứu cơ chế truy vấn các đối tượng chuyển động”**

**DỰ ÁN: Phát triển dịch vụ ứng dụng công nghệ GPS trong quản lý, giám sát, điều phối và tối ưu hóa kế hoạch sử dụng phương tiện**

**Mã số:...**

|  |  |
| --- | --- |
| **Cơ quan chủ trì dự án:** | **Đại học Công nghiệp Hà Nội** |
| **Chủ nhiệm dự án:** | **TS. Nguyễn Văn Tăng** |

**Người thực hiện chuyên đề: …**

**Hà Nội – 2016**

**Mục lục**

[Danh mục từ viết tắt 4](#_Toc461441154)

[Chương 1. Tổng quan về CSDL đối tượng chuyển động và các công nghệ 5](#_Toc461441155)

[1. Cơ sở dữ liệu đối tượng chuyển động 5](#_Toc461441156)

[a. Điểm chuyển động 6](#_Toc461441157)

[b. Vùng chuyển động 8](#_Toc461441158)

[c. Đặc tính của các đối tượng chuyển động 8](#_Toc461441159)

[2. Cơ sở dữ liệu không gian 9](#_Toc461441160)

[a. Mô phỏng khái niệm không gian 11](#_Toc461441161)

[b. Mở rộng mô hình dữ liệu và ngôn ngữ truy vấn 12](#_Toc461441162)

[3. Cơ sở dữ liệu thời gian 13](#_Toc461441163)

[a. Quản lý thời gian trong cơ sở dữ liệu dạng chuẩn 13](#_Toc461441164)

[b. Miền thời gian 14](#_Toc461441165)

[4. Dịch vụ công nghệ điện toán đám may Google App Engine (GAE) 14](#_Toc461441166)

[a. Môi trường ứng dụng: 15](#_Toc461441167)

[b. RunTimes và Frameworks: 16](#_Toc461441168)

[c. Hỗ trợ và độ tin cậy 16](#_Toc461441169)

[d. Google Cloud SQL 16](#_Toc461441170)

[e. Giới hạn sử dụng 17](#_Toc461441171)

[f. Free Quota 17](#_Toc461441172)

[5. Hệ thống định vị toàn cầu GPS – Global Positioning System 17](#_Toc461441173)

[a. Phần không gian 18](#_Toc461441174)

[b. Phần kiểm soát 19](#_Toc461441175)

[c. Phần sử dụng 19](#_Toc461441176)

[6. Kết luận 19](#_Toc461441177)

[Chương 2. Hệ thống lưu trữ CSDL đối tượng chuyển động 20](#_Toc461441178)

[1. Tổng quan về hệ thống lưu trữ CSDL đối tượng chuyển động 20](#_Toc461441179)

[2. Hệ thống lưu trữ CSDL đối tượng chuyển động Secondo 20](#_Toc461441180)

[a. Giới thiệu 20](#_Toc461441181)

[b. Kiến trúc 21](#_Toc461441182)

[c. Quản lý dữ liệu đối tượng chuyển động 26](#_Toc461441183)

[d. Truy vấn dữ liệu 27](#_Toc461441184)

[3. Kết luận 28](#_Toc461441185)

[Chương 3. Thiết kế và thử nghiệm trên hệ thống lưu trữ dữ liệu chuyển động 29](#_Toc461441186)

[1. Truy vấn 1 29](#_Toc461441187)

[2. Truy vấn 2 30](#_Toc461441188)

[3. Truy vấn 3 31](#_Toc461441189)

[4. Truy vấn 4 32](#_Toc461441190)

[5. Truy vấn 5 33](#_Toc461441191)

[6. Truy vấn 6 34](#_Toc461441192)

[Kết luận 36](#_Toc461441193)

# Danh mục từ viết tắt

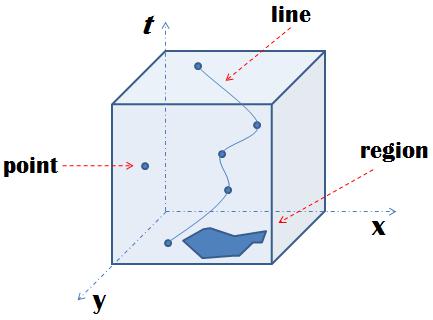
|  |  |
| --- | --- |
| API | Application Programming Interface |
| CSDL | Cơ sở dữ liệu |
| DBMS | Database Management System |
| GAE | Google App Engine |
| GPS | Global Position System |
| KTG | Không gian-thời gian (Spatio-temporal) |
| LBS | Location-based service |
| MOD | Moving Objects Databases |
| GUI | Graphic User Interface |

# Chương 1. Tổng quan về CSDL đối tượng chuyển động và các công nghệ

## Cơ sở dữ liệu đối tượng chuyển động

Cơ sở dữ liệu là tập hợp các dữ liệu có liên quan với nhau, được lưu trữ trên máy tính, có nhiều người sử dụng và được tổ chức theo một mô hình. Dữ liệu là những sự kiện có thể ghi lại được và có ý nghĩa. Tất cả các CSDL đều có thể được biểu diễn bởi hệ thống các thực thể, thuộc tính và mối quan hệ giữa các thực thể.

Cơ sở dữ liệu (CSDL) đối tượng chuyển động (Moving Objects Databases-MOD) là CSDL hỗ trợ cho mô hình và truy vấn của các dữ liệu địa lý phụ thuộc thời gian và các đối tượng chuyển động. Các ứng dụng dịch vụ dựa trên vị trí (Location based services - LBS) mà MOD cung cấp rất đa dạng và phong phú như các lĩnh vực: Quản lý và phân tích giao thông, phân tích hành vi người (khách hàng), nghiên cứu môi trường, sinh học (ví dụ theo dõi sự di cư của động vật theo mùa), khoa học Trái đất, lịch sử,…Xây dựng mô hình dữ liệu cho các đối tượng trong các ứng dụng đó là những yêu cầu luôn được đặt ra, có ba kiểu trừu tượng hóa cơ bản như hình 1 có thể được sử dụng: Điểm (*point*), đường (*line*), và vùng (region). Các đối tượng chuyển động đó cần phải được lưu trữ trong một CSDL.

****

*Hình 1. Kiểu dữ liệu không gian (point, line, region)*

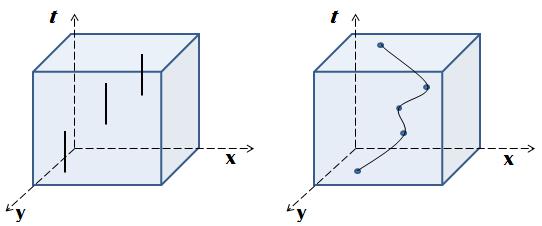
MOD đã được nghiên cứu trong khoảng 15 năm nay, các công trình nghiên cứu cơ sở dữ liệu MOD tập trung chủ yếu vào hai góc nhìn khác nhau như sau:

* **Góc nhìn quản lý vị trí** xem xét đến việc lưu trữtrong CSDL các vịtríhiện tại của các đối tượng chuyển động và các truy vấn cho các vị trí ở thời điểm hiện tại và tương lai gần. Cách tiếp cận này không quan tâm đến tính động của chuyển động vì nó không ghi lại tất cả lịch sử của đối tượng chuyển động. Nó được coi như một CSDL được ghi lại ngay, duy trì trạng thái hiện tại.
* **Góc nhìn dữ liệu không gian-thời gian** (KTG) xem xét việc lưu trữtrong một cơ sở dữ liệu không gian của dữ liệu chuyển động, với một quan tâm đến quan sát các sự thay đổi theo thời gian để phân tích hành vi của các đối tượng chuyển động từ dữ liệu lịch sử của chuyển động. Cách tiếp cận này giới thiệu hai kiểu đối tượng chuyển động trừu tượng cơ bản là điểm chuyển động (*moving point*) và vùng chuyển động (*moving region*). Đây là hai kiểu đối tượng chuyển động quan trọng nhất trong mô hình dữ liệu MOD và hai khái niệm này sẽ được làm rõ ở phần dưới đây.

### Điểm chuyển động

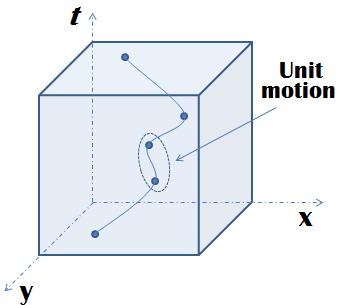
Một điểm chuyển động (*MPoint*) là trừu tượng cơ bản của một đối tượng vật lý chuyển động trên mặt phẳng hoặc trong không gian nhiều chiều, nó chỉ liên quan đến thay đổi vị trí. *MPoint* được định nghĩa khác như là một hàm liên tục từ thời gian vào không gian 2 chiều (2D), hoặc như một đường gấp khúc (*polyline*) trong không gian ba chiều (2D + thời gian). Khi phân tích hành vi chuyển động của người và động vật trong không gian, các chuyển động có thể dễ dàng được mô hình hóa và xử lý như một chuỗi các quan sát điểm chuyển động, được biểu diễn như là 3 bộ thời gian *t* và tọa độ (x, y). Ví dụ cho các đối tượng điểm chuyển động như: ô tô, máy bay, tàu chở hàng, người sử dụng điện thoại di động, động vật, v.v.

Các công trình nghiên cứu CSDL KTG giới thiệu sự phân loại của các dữ liệu điểm phụ thuộc vào thời gian. Những dữ liệu này có thể được xem một cách tự nhiên như được nhúng vào một không gian, đó là tích vectơ của miền không gian và thời gian và được trình diễn trong một không gian 3D như hình 2. Trong miền thời gian *t*, thời gian có thể được xem là rời rạc, dày đặc, hoặc liên tục, vì những lý do thực tế các mô hình CSDL thời gian (*temporal database*) thường dùng các thể hiện rời rạc của thời gian. Ngược lại, các mô hình liên tục là tương thích hơn để nói về với các đối tượng chuyển động.



*Hình 2. Điểm thay đổi rời rạc (trái) – Điểm thay đổi liên tục (phải)*

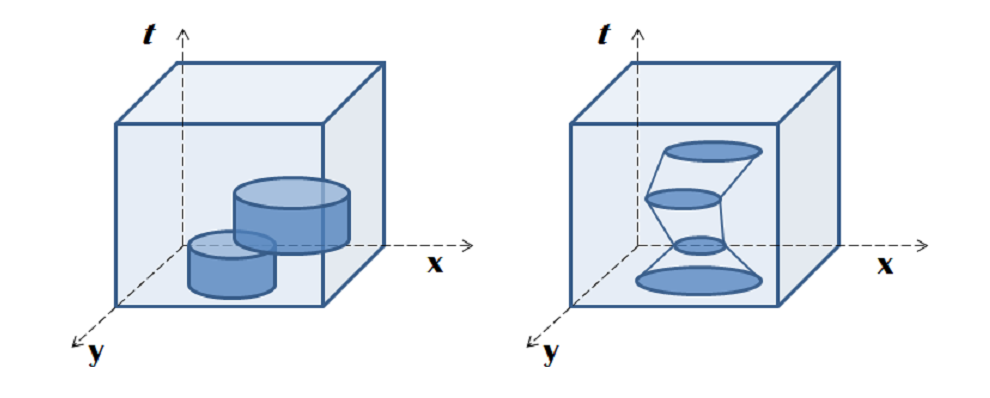
Các mô hình chuyển động của các điểm được tập trung vào các khái niệm về hành trình (*trajectory*), các dữ liệu chuyển động thu được bằng cách sử dụng bộ phần mềm hỗ trợ định vị và ghi lại các vị trí chuyển động của đối tượng chuyển động. Trình diễn cho sự chuyển động của một điểm chuyển động trong một hình thức rời rạc. Tiếp cận này thể hiện của thời gian liên kết với một đơn vị chuyển động của một điểm chuyển động trong mặt phẳng x, y như là một hàm của thời gian trong hình 3.



*Hình 3. Trình diễn rời rạc của một điểm chuyển động*

### Vùng chuyển động

Vùng chuyển động (*MRegion*) là trừu tượng cơ bản của một đối tương vật lý di chuyển xung quanh trong mặt phẳng hoặc trong một không gian nhiều chiều. Những MRegion là những thực thể chuyển động với sự thay đổi về độ lớn và vị trí theo thời gian trong không gian. Các ví dụ về *MRegion* như: cơn bão, cháy rừng, sự cố tràn dầu, sự lây lan của thảm thực vật hoặc của một căn bệnh, dịch bệnh, ...*MRegion* được trình diễn trong không gian 3 chiều như hình 1.4.



*Hình 4. Vùng thay đổi rời rạc (trái) – Vùng thay đổi liên tục (phải)*

### Đặc tính của các đối tượng chuyển động

Trong phần này, báo cáo trình bày một số đặc điểm của đối tượng chuyển động dựa trên các nghiên cứu của các nhà khoa học trong về việc lưu trữ, xử lý các mô hình và các truy vấn của dữ liệu chuyển động. Dữ liệu các đối tượng chuyển động thường được lưu trữ trong hệ thống cơ sở dữ liệu không gian - thời gian, cung cấp những ngôn ngữ cho việc đặt ra câu truy vấn. Các đặc tính của đối tượng chuyển động, đối với "hình dạng" của nó trong không gian 3D, có thể được tóm tắt như sau:

**Điểm**:

Các sự kiện trong không gian và thời gian: *<point, instant>;*

Địa điểm có giá trị trong một thời gian nhất định: *<point, period>*;

Thiết lập các vị trí sự kiện theo trình tự: *<point, instant>*;

Từng bước đổi vị trí trình tự: *<point, period>;*

Các đơn vị chuyển động: điểm chuyển động.

**Vùng**:

Các sự kiện khu vực trong không gian và thời gian: *<region,* *instant>*;

Khu vực có giá trị trong một khoảng thời gian: *<region, period>;*

Thiết lập các sự kiện khu vực: - chuỗi các *<region, instant>;*

Các vùng theo từng bước liên tục: - chuỗi các *<region, period>;*

Chuyển động các thực thể với mức độ: vùng chuyển động.

Một trong những ứng dụng của đối tượng chuyển động là ta có thể thực hiện các ứng dụng giám sát để mô hình hóa và truy vấn đối tượng chuyển động, sử dụng các phương pháp và các kỹ thuật như việc thực hiện các mô hình dữ liệu được cho bởi các modun đại số, mà cung cấp một tập hợp các xây dựng kiểu và một thiết lập của các nhà khai thác, phát triển. Các lĩnh vực chuyển động các đối tượng cơ sở dữ liệu xuất hiện đặc trưng bởi hai góc nhìn, quản lý vị trí và cơ sở dữ liệu KTG, như đã đề cập ở trên.

## Cơ sở dữ liệu không gian

Mục tiêu của nghiên cứu cơ sở dữ liệu không gian là mở rộng mô hình dữ liệu DBMS và ngôn ngữ truy vấn để có thể hiển thị và truy vấn hình học trong đường đi tự nhiên. Triển khai DBMS cần được gia hạn tương ứng cấu trúc dữ liệu cho nhiều kiểu dáng hình học, thuật toán để thực hiện kỹ thuật oán hình học, chỉ mục cho không gian đa chiều, và mở rộng tối ưu hóa (quy tắc dịch thuật, chức năng chi phí) để ánh xạ tới từ ngôn ngữ truy vấn đến mới thành phần liên quan đến hình học. Những động lực chính cho nghiên cứu cơ sở dữ liệu không gian là những hệ thống thông tin địa lý (*GIS - Geographic Information System*). Hệ thống GIS đầu đã chỉ sử dụng giới hạn của công nghệ DBMS, ví dụ, bằng cách lưu trữ dữ liệu phi không gian trong một DBMS như việc quản lý hình học một cách riêng biệt trong các tập tin. Dù thế, công nghệ cơ sở dữ liệu không gian đã phát triển như vậy mà bây giờ tất cả các nhà cung cấp DBMS chính (ví dụ như Oracle, IBM DB2, Informix) đề nghị mở rộng không gian. Vì vậy nó là dễ dàng hơn bây giờ để xây dựng GIS hoàn toàn như một lớp trên đầu trang của một DBMS, ví dụ, lưu trữ tất cả các dữ liệu trong các DBMS.

Thực tế, cơ sở dữ liệu không gian có phạm vi rộng lớn hơn. Ngoài không gian địa lý, có những không gian khác cần quan tâm có thể hiển thị trong cơ sở dữ liệu như là:

* Bố cục của thiết kế mạch tích hợp với quy mô lớn (là mảng lớn tập hợp các hình chữ nhật).
* 3D mô hình cơ thể người.
* Cấu trúc protein nghiên cứu trong sinh học phân tử.

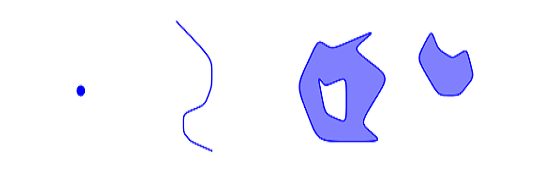
Một khác biệt quan trọng liên quan đến cơ sở dữ liệu không gian và thời gian. Mặc dầu không gian địa lý có thể được đại diện bởi những hình ảnh thu được bằng hình ảnh trên không hoặc vệ tinh, trọng tâm của DBMS không gian là đại diện cho các thực thể trong không gian với một định nghĩa rõ ràng vị trí và mức độ. Cơ sở dữ liệu hình ảnh quản lý hình ảnh như vậy. Tất nhiên, có tồn tại kết nối. Ví dụ, các kỹ thuật khai thác tính năng có thể được sử dụng để xác định trong vòng một hình ảnh thực thể không gian có thể được lưu trữ trong cơ sở dữ liệu không gian.

### Mô phỏng khái niệm không gian

Những thực thể để được lưu trữ trong cơ sở dữ liệu không gian? Nếu xét là dấu cách địa lý, bất cứ thứ gì rõ ràng có đủ khả năng có thể xuất hiện trong bản đồ giấy, chẳng hạn, thành phố, sông, mạng đường cao tốc, điểm mốc, ranh giới của quốc gia, bệnh viện, ga tàu điện ngầm, rừng, cánh đồng bắp, và những thứ tương tự như thế.

Để mô hình thực thể muôn hình vạn trạng, một này có thể đưa ra khái niệm để mô hình đối tượng đơn và không gian liên quan tập hợp các đối tượng.

Để mô phỏng vật thể đơn, ba khái niệm trừu tượng cơ bản là điểm, đường, và vùng. Thể hiện điểm (khía cạnh hình học) của vật thể, dành riêng cho mà nó vị trí trong không gian, nhưng nó không phải mức độ, có liên quan. Ví dụ vật thể điểm là thành phố trên quy mô lớn bản đồ, điểm mốc, bệnh viện, hoặc ga tàu điện ngầm. Đường (đường trong không gian) là khái niệm trừu tượng cơ bản cho di chuyển qua không gian, hoặc kết nối trong không gian. Ví dụ vật thể đường là sông, đường cao tốc, hoặc cáp điện thoại. Cuối cùng, vùng là khái niệm trừu tượng cho thực thể có mức độ trong không gian 2D. Vùng có thể nói chung có lỗ và bao gồm nhiều mảnh rời rạc. Ví dụ vật thể vùng là quốc gia, rừng, hoặc hồ. Ba khái niệm trừu tượng cơ bản được minh hoạ trong hình 5.



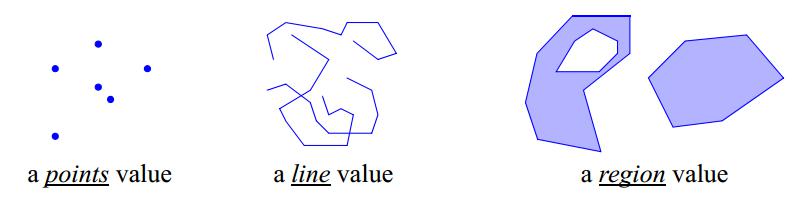
*Hình 5. Điểm, đường và vùng*

### Mở rộng mô hình dữ liệu và ngôn ngữ truy vấn

Hiện nay ta xem xét cách khái niệm trừu tượng cơ bản có thể nhúng vào mô hình dữ liệu DBMS.

Cho đơn khái niệm trừu tượng đối tượng đơn, đường, và vùng, đó là điều tự nhiên để giới thiệu tương ứng kiểu dữ liệu trừu tượng, hoặc kiểu dữ liệu không gian (spatial data types - SDTs). Chẳng hạn như thử nghiệm xem hai vùng là đang liền kề hoặc một được bao bởi vùng khác, hoặc như tạo thành sự khác biệt của hai vùng hoặc giao lộ của đường với vùng hay khoảng cách giữa điểm và đường, hoặc tập hợp các vùng vào một vùng, hoặc tìm trong tập hợp các điểm đến điểm truy vấn.

Tập hợp các kiểu dữ liệu không gian với thao tác liên quan tạo thành đại số không gian. Chẳng hạn như, giao lộ của hai đường giá trị sinh ra nói chung tập hợp các điểm, và sự khác biệt của hai vùng, dù là mỗi đối số là vùng đơn giản không có lỗ, có thể mang lại một vùng gồm vài thành phần rời rạc có lỗ. Các kiểu dữ liệu được gọi là điểm, đường, và vùng có cấu trúc là minh họa trong hình 6.



*Hình 6. Các kiểu dữ liệu không gian điểm, đường và vùng*

Ở đây kiểu *points* cung cấp tập hợp các điểm, kiểu *line* tập hợp các đường, và kiểu *region* tập hợp các hình đa giác với lỗ. Dữ liệu từng kiểu không gian được định nghĩa, chúng có thể nhúng vào mô hình dữ liệu DBMS trong vai trò của kiểu thuộc tính. Vậy nên ngoài kiểu chuẩn như là *int, real, string,* có thể có kiểu không gian *points, line,* và *region.* Kiểu này có thể được dùng trong bất kỳkiểu mô hình dữliệu DBMS;dù nó là quan hệ, hướng đối tượng, hoặc một cái gì đó nữa chẳng quan trọng.

## Cơ sở dữ liệu thời gian

### Quản lý thời gian trong cơ sở dữ liệu dạng chuẩn

Cơ sở dữ liệu được quản lý bởi chuẩn DBMS thường mô tả trạng thái hiện hành của thế giới theo như khi nó được biết ở cơ sở dữ liệu. Thay đổi tình trạng hiện tại của thế giới sẽ được phản ánh muộn một chút rồi mới cập nhật đến cơ sở dữ liệu sau khi trạng thái trước đây mất đi. Tất nhiên, nhiều ứng dụng của nó là không đủ để chỉ duy trì trạng thái hiện hành; chúng cần theo dõi lịch sử một đoạn đường đi. Ở DBMS chuẩn đây là có thể nếu ứng dụng quản lý thời gian chính nó, bằng thêm thuộc tính thời gian rõ ràng và thực hiện đúng kiểu tính toán trong truy vấn. Chẳng hạn như, giả sử công ty có bảng nhân viên với mẫu:

|  |
| --- |
| **NHANVIEN (name: string, department: string, salary: int)** |

Nếu công ty muốn theo dõi bộ phận và lƣơng trước đây của nhân viên, bảng có thể được mở rộng:

|  |
| --- |
| **NHANVIEN (name: string, department: string, salary: int, start: date, end: date)** |

DBMS chuẩn đưa ra hạn chế rất hỗ trợ này dưới dạng kiểu dữ liệu như là ngày hoặc thời gian. Tuy nhiên, xử lý thời gian trong mẫu đơn này bằng ứng dụng là khó khăn, dễ mắc lỗi, dẫn đến thành lập công thức truy vấn phức tạp và thường thi hành truy vấn không hiệu quả. Chẳng hạn như, trong chỗ nối của hai bảng mở rộng thuộc tính thời gian ở trên, cần phải bảo đảm thứ tự khoảng thời gian được nối lại tránh chồng lấp, bằng cách thêm điều kiện rõ ràng trong truy vấn. Điều kiện này đem lại vài sự bất bình đẳng về thuộc tính thời gian. Chuẩn DBMS thường không tốt lắm khi xử lý sự bất bình đẳng trong tối ưu hoá truy vấn, vậy nên thi hành có thể không hiệu quả. Ngược lại, nếu hỗ trợ thời gian đúng được đưa vào DBMS, việc này có thể được thực hiện tự động; tình trạng không được cần đến trong truy vấn, và thi hành sẽ được điều chỉnh để thực hiện kiểu liên kết này rất hữu hiệu.

Vậy nên mục tiêu của nghiên cứu cơ sở dữ liệu thời gian đã từng đến tích hợp khái niệm thời gian sâu vào DBMS mô hình dữ liệu và ngôn ngữ truy vấn và để mở rộng hệ thống thích hợp để đạt được thi hành hiệu quả.

### Miền thời gian

Trước tiên, hãy xem xét bao nhiêu thời gian bản thân có thể mô hình. Thời gian nói chung được nhận thấy như không gian một chiều mở rộng từ quá khứ, đến tương lai. Có một số tùy chọn:

* Khoảng thời gian có thể được xem là giới hạn hoặc vô tận. Mô hình giới hạn thừa nhận một số nguồn gốc và cũng là điểm cuối của thời gian.
* Thời gian có thể được xem là riêng rẽ, dày đặc, hoặc liên tục *(discrete,* *dense, or continuous.).* Mô hình riêng rẽlàđẳng cấuđến sốtựnhiên hoặc sốnguyên. Mô hình dày đặc đẳng cấu để cả hai hoặc số hữu tỉ hoặc số thực: giữa hai thời điểm của thời gian vừa tồn tại. Mô hình liên tục là đẳng cấu đến số thực. Xét vì hầu hết mọi người sẽ nhận thấy thời gian khi được liên tục, vì lý do thực tiễn mô hình cơ sở dữ liệu thời gian thường sử dụng biểu diễn sự rời rạc của thời gian. Ngược lại, sau này này mô hình tiến trình liên tục sẽ được dùng, vì này thích hợp hơn để xử lý đối tượng chuyển động.

## Dịch vụ công nghệ điện toán đám may Google App Engine (GAE)

Google App Engine (thường được gọi tắt là GAE hoặc App Engine) là nền tảng PaaS được Google cung cấp nhằm phát triển các ứng dụng web và được lưu trữ tại Trung tâm quản lý dữ liệu Google (Google-managed data center). Các ứng dụng được đóng gói và chạy trên các máy chủ.

Google App Engine cho phép người dùng chạy các ứng dụng web trên nền tảng kiến trúc Google. Các ứng dụng App Engine được xây dựng và quản lý một cách dễ dàng. Với App Engine, không cần phải bào trì máy chủ, người dùng upload các ứng dụng và nó luôn sẵn sàng phục vụ.

App Engine tự động tổng hợp yêu cầu cho các ứng dụng, như: tự động tổng hợp các yêu cầu của ứng dụng, tự động xác định vị trí các tài nguyên cần thiết để ứng dụng có thể đáp ứng các yêu cầu người dùng.

Người dùng có thể sử dụng domain được cung cấp bởi Google Apps hoặc sử dụng domain riêng cho các ứng dụng của mình. Các ứng dụng có thể được chia sẻ rộng rãi hoặc giới hạn cho một nhóm người dùng nhất định.

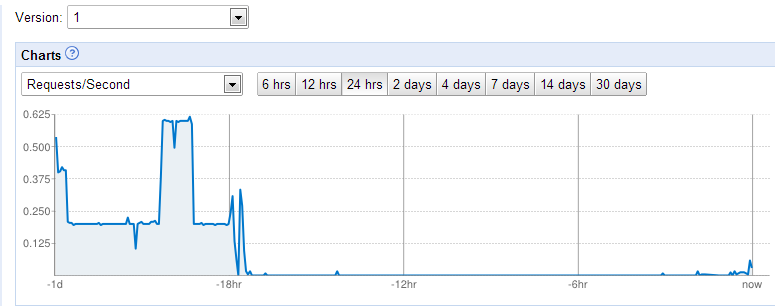
Khi sử dụng, Google App Engine miễn phí sử dụng tài nguyên, chỉ tính phí băng thông, mở rộng lưu trữ. Hay nói cách khác, người dùng chỉ phải trả phí cho những tài nguyên sử dụng. Không có phí thuê bao hay các khoản phụ phí khác. Các tài nguyên được sử dụng như dung lượng lưu trữ, băng thông… đều được quản lý sử dụng với hiệu quả tối đa cho người dùng.

Google App Engine được ra mắt lần đầu tiên vào 08/2008.

### Môi trường ứng dụng:

App Engine cung cấp môi trường cho phép xây dựng ứng dụng đơn giản nhưng hoạt động có độ tin cậy cao. Các đặc điểm của môi trường ứng dụng App Engine:

* Dịch vụ dynamic web, được hỗ trợ đầy đủ các công nghệ web hiện đại.
* Lưu trữ tin cậy hỗ trợ các truy vấn, sắp xếp, và tác vụ.
* Tự động điều chỉnh cân bằng tải nhằm tạo hiệu suất phục vụ cao nhất.
* API hỗ trợ xác thực người dùng và gửi thư điện tử bằng tài khoản Google.
* Bộ giả lập hoàn chỉnh hỗ trợ cho người dùng có thể phát triển ứng dụng và thử nghiệm trong môi trường ảo trên máy tính trước khi đưa lên mạng.
* Hỗ trợ các yêu cầu nằm ngoài giới hạn các web request.
* Hỗ trợ lịch trình hoạt động các tác vụ theo thời gian tự động.



*Hình 7. Bảng theo dõi hoạt động ứng dụng của GAE*

### RunTimes và Frameworks:

Hiện nay, Google App Engine hỗ trợ lập trình bằng ngôn ngữ Python, Java và Go.

* Python web framework đã hoạt động trên Google App Engine bằng cách tích hợp GAE vào Django, CherryPy, Pylons…
* Google App Engines hỗ trợ lập trình ứng dụng bằng ngôn ngữ Java. Với công nghệ lõi là servlet 2.5 technology được dùng trên web server mã nguồn mở Jetty Web Server, đi kèm với một số công nghệ như JSP.

### Hỗ trợ và độ tin cậy

Thời gian hoạt động của các ứng dụng là 99.95%

### Google Cloud SQL

Vào tháng 10/2011, Google giới thiệu dịch vụ CSDL, hỗ trợ JDBC và DB – API. Dịch vụ này cho phép người dùng Khởi tạo, Cấu hình, và sử dụng CSDL quan hệ trong các ứng dụng App Engine. Dịch vụ sử dụng engine MySQL Version 5.1.59 và kích thước CSDL tối đa không quá 10GB.

### Giới hạn sử dụng

Google App Engine cần phải có tài khoản Google để kích hoạt. Mỗi tài khoản cho phép đăng ký tối đa 10 ứng dụng. Giới hạn này có thể thay đổi theo chính sách của Google và đội ngũ kỹ thuật của Google.

Google App Engine cũng đặt ra giới hạn cho các ứng dụng miễn phí.

|  |  |
| --- | --- |
| **Quota** | **Limit** |
| Time per request | 60 sec per normal request, 10 minutes for tasks, unlimited for backends |
| HTTP response size | 32 MB |
| Datastore item size | 1 MB |
| Application code size | 150 MB |

*Hình 8. Bảng mô tả giới hạn sử dụng của một ứng dụng GAE miễn phí*

### Free Quota

Google App Engine có các chính sách hỗ trợ những người dùng mới. Khi mới bắt đầu sử dụng dịch vụ, người sử dụng sẽ không phải trả bất cứ chi phí nào trong một giới hạn cho phép gọi là free quota.

## Hệ thống định vị toàn cầu GPS – Global Positioning System

GPS - Hệ thống định vị toàn cầu là hệ thống xác định vị trí dựa trên vị trí của các vệ tinh nhân tạo, do Bộ Quốc phòng Hoa Kỳ thiết kế, xây dựng, vận hành và quản lý. Trong cùng một thời điểm, tọa độ của một điểm trên mặt đất sẽ được xác định nếu xác định được khoảng cách từ điểm đó đến ít nhất ba vệ tinh.

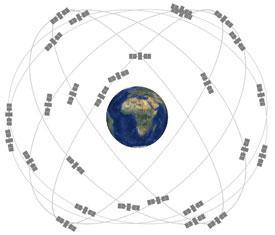
GPS sử dụng nguyên tắc hướng thẳng tương đối của hình học và lượng giác học. Mỗi vệ tinh liên tục phát và truyền dữ liệu trong quỹ đạo của nó, do đó, mỗi thiết bị GPS nhận sẽ liên tục truy cập dữ liệu quỹ đạo chính xác từ vị trí của tất cả vệ tinh. Từ đó tín hiệu hoặc sóng vô tuyến di chuyển ở vận tốc hằng số (thường bằng vận tốc ánh sáng – C), các thiết bị GPS thu có thể tính toán khoảng cách liên quan từ GPS đến các vệ tinh khác bằng cách máy thu GPS so sánh thời gian tín hiệu được phát đi từ vệ tinh với thời gian mà thiết bị GPS thu nhận được tín hiệu do các vệ tinh khác. Nguyên lý xác định toạ độ của hệ thống GPS dựa trên công thức quãng đường bằng vận tốc x thời gian. Vệ tinh phát ra các tín hiệu bao gồm vị trí của chúng, thời điểm phát tín hiệu.

Máy thu tính toán được khoảng cách từ các vệ tinh, giao điểm của các mặt cầu có tâm là các vệ tinh, bán kính là thời gian tín hiệu đi từ vệ tinh đến máy thu nhân vận tốc sóng điện từ là toạ độ điểm cần định vị.

GPS hiện tại gồm 3 phần chính: Phần không gian, phần kiểm soát và phần sử dụng.

### Phần không gian

Phần không gian gồm 27 vệ tinh (24 vệ tinh hoạt động và 3 vệ tinh dự phòng) nằm trên các quỹ đạo xoay quanh trái đất. Chúng cách mặt đất 20.200 km, bán kính quỹ đạo 26.600 km.



*Hình 9. Vệ tinh GPS*

Chúng chuyển động ổn định và quay hai vòng quỹ đạo trong khoảng thời gian gần 24 giờ với vận tốc 7 nghìn dặm một giờ. Các vệ tinh trên quỹ đạo được bố trí sao cho các máy thu GPS trên mặt đất có thể nhìn thấy tối thiểu 4 vệ tinh vào bất kỳ thời điểm nào.

Các vệ tinh được cung cấp bằng  [năng lượng Mặt Trời.](http://vi.wikipedia.org/wiki/N%C4%83ng_l%C6%B0%E1%BB%A3ng_M%E1%BA%B7t_Tr%E1%BB%9Di) Chúng có các nguồn  [pin](http://vi.wikipedia.org/wiki/Pin_(%C4%91%E1%BB%8Bnh_h%C6%B0%E1%BB%9Bng)) dự phòng để duy trì hoạt động khi chạy khuất vào vùng không có ánh sáng  [Mặt Trời.](http://vi.wikipedia.org/wiki/M%E1%BA%B7t_Tr%E1%BB%9Di) Các  [tên lửa](http://vi.wikipedia.org/wiki/T%C3%AAn_l%E1%BB%ADa) nhỏ gắn ở mỗi quả vệ tinh giữ chúng bay đúng quỹ đạo đã định.

### Phần kiểm soát

Mục đích trong phần này là kiểm soát vệ tinh đi đúng hướng theo quỹ đạo và thông tin thời gian chính xác. Có 5 trạm kiểm soát đặt rải rác trên trái đất. Bốn trạm kiểm soát hoạt động một cách tự động, và một trạm kiểm soát là trung tâm. Bốn trạm này nhận tín hiệu liên tục từ những vệ tinh và gửi các thông tin này đến trạm kiểm soát trung tâm. Tại trạm kiểm soát trung tâm, nó sẽ sửa lại dữ liệu cho đúng và kết hợp với hai ăng-ten khác để gửi lại thông tin cho các vệ tinh. Ngoài ra, còn một trạm kiểm soát trung tâm dự phòng và sáu trạm quan sát chuyên biệt.

### Phần sử dụng

Phần sử dụng là thiết bị nhận tín hiệu vệ tinh GPS và người sử dụng thiết bị này.

## Kết luận

Chương 1 trình bày tổng quan về CSDL đối tượng chuyển động và vai trò của nó trong các lĩnh vực của cuộc sống. Trình bày về hai dạng CSDL không gian, thời gian, về hai góc nhìn cùng với các đối tượng chuyển động quan trọng nhất của mô hình MOD.

Đồng thời cũng trình bày sơ lược về dịch vụ công nghệ điện toán đám mây, hệ thống GPS... là những công cụ hỗ trợ cho việc thử nghiệm ứng dụng.

# Chương 2. Hệ thống lưu trữ CSDL đối tượng chuyển động

## Tổng quan về hệ thống lưu trữ CSDL đối tượng chuyển động

Sự bùng nổ của các thiết bị di động và máy tính hiệu năng cao đã tác động không nhỏ đến đời sống của con người. Thay vì các cơ sở dữ liệu (CSDL) truyền thống lưu trữ các dữ liệu tĩnh (hay còn gọi là dữ liệu phi không gian và thời gian) về các đối tượng như họ tên, ngày tháng, nghề nghiệp…, xu hướng xây dựng các hệ thống CSDL qui mô cực lớn liên quan tới các thông tin thời gian và không gian (thậm chí có yếu tố thời gian thực) đang được triển khai mạnh mẽ trên thế giới. Cụ thể, các CSDL của đối tượng chuyển động được lưu trữ trong các hệ thống CSDL không gian-thời gian lớn được mở rộng theo cấp số nhân. Các hệ thống này đang được triển khai và hỗ trợ đắc lực cho các nghiệp vụ quản lý (ví dụ: quản lý lưu lượng giao thông đô thị, lập dự án xây đường cao tốc để giảm tải…) và nghiên cứu khoa học (ví dụ: Sự di cư của các loài động vật theo mùa; di dân giữa các quốc gia, thành phố…).

## Hệ thống lưu trữ CSDL đối tượng chuyển động Secondo

### Giới thiệu

Secondo là một hệ thống CSDL mở rộng, một DBMS thử nghiệm đã được phát triển tại Đại học Hagen (Đức) từ khoảng năm 1995. Nó chạy trên các hệ điều hành Windows, Linux, Mac-OS X và là phần mềm nguồn mở [8]. Các mục tiêu thiết kế chính là một kiến trúc mở rộng tiện ích và hỗ trợ cho những ứng dụng không gian và không gian - thời gian.

Mục tiêu của Secondo để cung cấp một mô hình chung cho các hệ thống cơ sở dữ liệu cùng với những công cụ khai thác của nhiều mô hình cơ sở dữ liệu khác nhau. Chẳng hạn như có thể để triển khai mô hình cơ sở dữ liệu quan hệ, cơ sở dữ liệu hướng đối tượng, thời gian, hoặc mô hình XML và có thể hỗ trợ các kiểu dữ liệu về không gian, đối tượng chuyển động, công thức hoá học, v.v. Trong khi khả năng mở rộng bằng kiểu dữ liệu phổ biến hiện nay còn nhiều hạn chế, thì khả năng thay đổi mô hình dữ liệu được thực hiện khá dễ dàng trong Secondo.

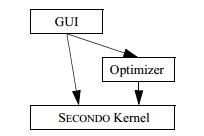
Để có được các ưu điểm đó hệ thống Secondo đã thực hiện:

* Riêng biệt mô hình dữ liệu độc lập thành phần và cơ chế trong hệ thống cơ sở dữ liệu (khung hệ thống) từ mô hình dữ liệu phần phụ thuộc.
* Cung cấp một hình thức để mô tả các mô hình dữ liệu thực hiện, để có thể cung cấp giao diện sạch giữa khung hệ thống và "nội dung". Hình thức này là chữ ký thứ hai-trật tự, giải thích dưới đây.
* Cơ cấu tổ chức thực hiện một mô hình dữ liệu vào một tập hợp các mô đun đại số, mỗi cung cấp các cấu trúc dữ liệu cụ thể và các hoạt động.

Secondo đã được định ban đầu như nền tảng cho triển khai và thí nghiệm mới kiểu mô hình dữ liệu, đặc biệt để hỗ trợ dữ liệu không gian, dữ liệu thời gian, và mô hình cơ sở dữ liệu đồ thị. Hiện nay Secondo có kiến trúc rõ ràng, và tạo được sự cân bằng hợp lý giữa tinh giản hoá và tinh tế.

### Kiến trúc

Kiến trúc của Secondo (tham khảo Hình 10) bao gồm ba thành phần chính là: Nhân (kernel), tối ưu hóa truy vấn (optimizer) và giao diện người dùng (GUI). Cơ chế làm việc của SECONDO là người dùng nhập câu truy vấn vào GUI:



*Hình 10. Kiến trúc hệ thống Secondo*

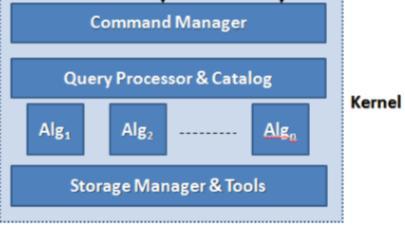
#### **Giao diện người dùng (GUI)**

GUI gửi các truy vấn tới Kernel hoặc tới Optimizer (Optimizer xử lý các truy vấn và gửi tới Kernel) và trình diễn kết quả truy vấn. GUI được viết bằng ngôn ngữ Java.

#### **Nhân Secondo (Secondo Kernel)**

Phần nhân được cấu tạo lên bởi các mô hình dữ liệu khác nhau (dạng truyền thống và các mô hình dữ liệu phục vụ việc nâng cấp độ thêm chiều không gian (GIS) và thời gian) cùng với các phương thức thao tác, ràng buộc giữa chúng. Bên cạnh đó, phần nhân còn có thành phần xử lý, tối ưu hóa câu lệnh truy vấn.

Mô tả sơ lược về kiến trúc của nhân SECONDO được trình bày trong hình 11. Mô hình dữ liệu là công cụ như những kiểu dữ liệu tập hợp và những thao tác. Chúng được phân thành nhiều nhóm đại số.



*Hình 11. Kiến trúc thô của nhân trong hệ thống Secondo*

Định nghĩa của đại số được dựa trên khái niệm của chữ ký thứ hai. Ý tưởng là sử dụng hai chữ ký ghép. Chữ ký nào cũng được cung cấp các phân loại và các thao tác. Ở đây trong chữ ký đầu tiên có thứ tự được gọi là *kiểu* và đại diện cho tập hợp các *kiểu*. Thao tác này của chữ ký là xây dựng kiểu. Chữ ký định nghĩa làm thế nào xây dựng kiểu và có thể được áp dụng cho những kiểu nào. Các kiểu có sẵn trong hệ thống được cho là chính xác của chữ ký này.

Chữ ký thứ hai định nghĩa thao tác trên kiểu của chữ ký đầu tiên. Mô đun đại số cung cấp tập hợp các xây dựng kiểu, triển khai cấu trúc dữ liệu đối với mỗi chúng. Một tập hợp nhỏ các chức năng hỗ trợ cần có để thanh ghi xây dựng kiểu trong đại số.

Tương tự, mô đun đại số đưa ra điều hành viên, triển khai chức năng hỗ trợ cho họ như là kiểu bản đồ, sự đánh giá, giải quyết sự quá tải, v.v. Bộ xử lý truy vấn đánh giá truy vấn bằng cách xây dựng cây tìm kiếm, thực hiện các thuật toán trên cây.

Trong Secondo phần nhân quản lý cơ sở dữ liệu. Cơ sở dữ liệu là tập hợp các đối tượng. Mỗi đối tượng gồm ba thuộc tính (*name, type, value*). Cơ sở dữ liệu có thể được tạo lập, xoá bỏ, mở, đóng, xuất dữ liệu vào và nhập dữ liệu từ các tập tin. Trong tập tin chúng được biểu diễn như danh sách lồng nhau dưới dạng thuần văn bản.

Trên cơ sở dữ liệu mở, có một số lệnh cơ bản sẵn có:

|  |
| --- |
| *type <ident> = <type expr>*  *delete type <ident>*  *create <ident>: <type expr>*  *update <ident> := <value*  *expr> let <ident> = <value*  *expr> delete <ident>*  *query <value expr>* |

Rõ ràng, kiểu biểu thức và giá trị biểu thức được định nghĩa trên đại số quan hệ. Phần nhân đưa ra các lệnh thêm cho hệ thống triển khai *(list type* *constructors, list operators, list algebras, list algebra <algebraName>),* chocơ sở dữ liệu sẵn có *(list databases),* hoặc nội dung của cơ sở dữ liệu mở *(list* *types, list objects)*.Đối tượng cũng có thểxuất ra và phục hồi từtập tin. Cuốicùng có lệnh cho quản lý giao dịch.

#### **Tối ưu hóa (Optimizer)**

Secondo là môi trường DBMS dễ mở rộng cho nghiên cứu việc thiết kế mẫu và công việc giảng dạy. Lẽ ra ta có thể thêm đại số mà không cần phải nỗ lực quá nhiều, đặc biệt với các đại số chuẩn hóa kiểu dữ liệu và thao tác, như đại số không gian hoặc đại số cho đối tượng di chuyển.

* Tương tự như các phương pháp tối ưu hóa khác, nó cần phải tìm được cách truy vấn cho kết quả nhanh.
* Cho phép làm việc với nhiều kiểu dữ liệu không chuẩn bằng nhiều thao tác. Trong thực tế, đưa ra được phương pháp lựa chọn kiểu dữ liệu này trong môi trường là một vấn đề lớn.
* Cài đặt các thuật toán ưu hoá cho dữ liệu không chuẩn.
* Cơ chế phần mở rộng cho tối ưu hóa phải đơn giản và dễ hiểu.
* Cho công việc giảng dạy, nó là ưu việt nếu thuật toán tối ưu hoá là đơn giản và rõ, và mã của tối ưu hóa dễ dàng hiểu được và có đầy đủ tài liệu dẫn chứng.

Optimizer được viết bằng PROLOG, chạy trong môi trường SWI-PROLOG (SWI04). SWI-PROLOG hỗ trợ ba nền tảng Windows, Linux and Solaris, giống như SECONDO.

PROLOG là ngôn ngữ xuất sắc cho triển khai bộ tối ưu truy vấn. So sánh mẫu và khả năng tìm kiếm của PROLOG làm thành lập công thức của quy tắc tối ưu hoá khá dễ và thi hành hoàn toàn nhanh. Sơ đồ liên kết truy vấn có đến 9 thuộc tính được xác định ngay; từ 10 trở đi, thời gian tối ưu hoá trở thành đáng chú ý.

Công cụ PROLOG rất được điều chỉnh cho công việc như vậy; thi hành tối ưu hóa này nhanh hơn nhiều so với bất cứ thứ gì có trong C + +.

Hiện tất cả đây là được dành riêng cho môi trường quan hệ. Tối ưu hoá truy vấn liên kết nghĩa là tối ưu hóa được cho trước tập hợp các quan hệ và tập hợp các lựa chọn và liên kết khẳng định và tạo ra sơ đồ có thể được thi hành bằng Secondo.

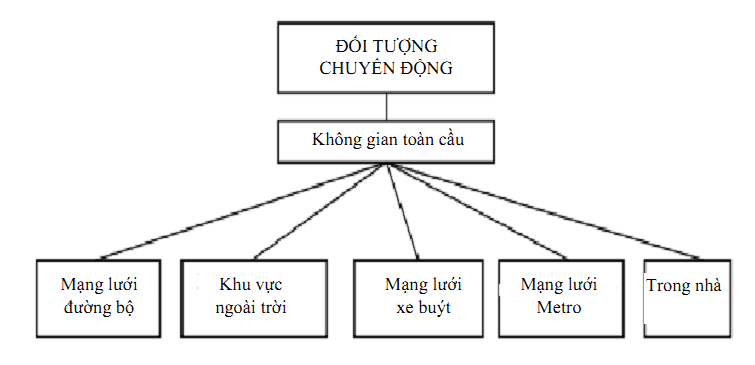
Ngôn ngữ truy vấn là SQLLite, trong ký hiệu thích ứng với PROLOG sao cho truy vấn có thể nhập trực tiếp trên giao diện PROLOG, nếu muốn.

Vì thế, trái với những cái khác của Secondo, tối ưu hóa không phải là mô hình dữ liệu độc lập. Đặc biệt, truy vấn bị hạn chế bởi cấu trúc của SQL và cũng là sự thật phải thừa nhận về mô hình quan hệ. Mặt khác, khả năng nòng cốt của tối ưu hóa để cho truy vấn liên kết có hiệu quả là cần đến ở bất kỳ loại mô hình dữ liệu.

Tối ưu hóa làm việc tuyệt đối với tư cách một phần độc lập. Nó truy cập phần nhân Secondo chỉ thông qua giao diện chuẩn của nó có thể được sử dụng bởi ứng dụng nào. Ký hiệu cho sơ đồ truy vấn là đại số triển khai bởi phần nhân; tất cả sơ đồ này có thể nhập bằng tay. Điều này có nghĩa là phần tối ưu hóa là rất dễ tích hợp vào các môi trường cung cấp xử lý truy vấn (với giao diện thân thiện với người dùng).

Như giao diện cao cấp nhất, tối ưu hóa cung cấp thuộc tính *optimize (Query, Plan, Cost)* mà có thể được gọi là từ môi trường khác, chẳnghạn như giao diện người dùng viết ở Java. Vì thế, đưa ra một truy vấn, nó trả về kết quả là một sơ đồ và ước tính chi phí.

### Quản lý dữ liệu đối tượng chuyển động



*Hình 12. Mô hình quản lý dữ liệu chuyển động*

Mô hình đối tượng chuyển động trong môi trường thế giới thực bao gồm: mạng lưới đường bộ, khu vực ngoài trời, mạng lưới xe buýt, mạng lưới metro, khu vực trong nhà. Đó là những môi trường tạo thành không gian cho đối tượng chuyển động. Sau đó, vị trí của đối tượng chuyển động được tham chiếu đến những không gian này. Do đó, việc quản lý dữ liệu được chia thành ba thành phần chính: Dữ liệu cơ sở hạ tầng; không gian lưu trữ; và các đối tượng chuyển động [5].

Môi trường (còn gọi là cơ sở hạ tầng) bao gồm một tập các đối tượng địa lý như hệ thống giao thông đường bộ, công trình giao thông… Những môi trường này đều có đặc thù riêng, các kiểu dữ liệu khác nhau của nó được đề xuất để đại diện cho các đối tượng. Ví dụ, một đường thẳng được sử dụng để mô tả con đường, một hình đa giác được dùng để biểu diễn công trình giao thông. Dữ liệu kiểu này được thiết kế theo các tính năng của các đối tượng khác nhau. Khi xem xét phạm vi trong nhà một tòa nhà bao gồm tập các phòng. Các kiểu dữ liệu cho một phòng nên được mô phỏng bằng hình ảnh 2D và chiều cao so với mặt đất. Các thông tin của môi trường được sử dụng để xác định vị trí cho các đối tượng chuyển động chung. Nó không chỉ dành cho các đối tượng tĩnh, mà còn có hệ thống xe buýt, các thành phố lớn nơi các đối tượng có thể di chuyển được.

Mỗi đối tượng địa lý được gán một ID duy nhất và một tập các dãy rời nhau được định nghĩa cho mọi môi trường, ví dụ [1, 5000] cho mạng lưới đường bộ, [6000, 7000] cho mạng lưới xe buýt. Các không gian được xây dựng trên đầu trang của các môi trường là nền tảng cho các đối tượng chuyển động chung chung và người ta có thể xác định môi trường địa điểm cho một ID của đối tượng địa lý nhất định. Sau đó, chúng được đại diện cho vị trí của đối tượng di chuyển bằng hai bộ phần được định nghĩa như sau:

*Dgenloc={(oid, (loc1, loc2)) | oid Dint , loc1, loc2 Dreal}*

### Truy vấn dữ liệu

Một nhóm các truy vấn về phương thức vận chuyển và mô trường di động có thể được xây dựng bằng một ngôn ngữ giống với SQL như bằng cách sử dụng đề xuất của nhà khai thác và các đối tượng giao diện quản lý quan hệ di chuyển và dữ liệu cơ sở hạ tầng [5]. Một ví dụ truy vấn được đưa ra dưới đây được thực hiện trong cơ sở dữ liệu Secondo:

* *Q1. Những ai đã ngồi trên xe buýt mang số hiệu 32 vào lúc 8 sáng thứ hai?* Truy vấn này đối với du khách đi xe buýt từ một tuyến cụ thể tại các thời điểm nhất định. Và tất cả các xe buýt hiện có di chuyển trên các tuyến đường tại các thời gian truy vấn.
* *Q2. Ai đã đến trường đại học bằng taxi vào thứ sáu?*
* *Q3. Ai đã lên chuyến xe buýt số 3 tại điểm dừng xe buýt tại trường đại học vào buổi chiều thứ ba?*

Để trả lời hai câu hỏi trên ta phải tìm nơi mà có người thay đổi phương thức duy chuyển.

* *Q4. Đã có ai sống ở tầng H5 của tòa nhà đi xe buýt đến ga xe lửa trong khoản thời gian 02:00-05:00 vào thứ sáu?*
* *Q5. Xe buýt số 35 có vƣợt qua bất kỳ xe đạp du lịch nào vào thứ hai không?*

Xem xét khoảng cách giữa hai vật thể di động với phương thức vận tải khác nhau. Khám phá mối liên hệ đó là thú vị khi hai đối tượng di chuyển trong môi trường khác nhau. Ở đây có nghĩa là tính tất cả các xe buýt mà mang số 35, nhưng với thời gian khởi hành khác nhau, thay vì tính một chiếc xe buýt đặc biệt.

* *Q6. Tìm hiểu tất cả mọi người ở tại phòng 154 trong trường đại học trong hơn một giờ đồng hồ vào hôm thứ năm.*

Bên cạnh những chuyến đi ngoài trời, các đối tượng di chuyển trong nhà cũng có thể là truy tìm.

* *Q7. Tìm các tuyến xe buýt k đầu với lưu lượng hành khách cao tất cả các ngày làm việc.*
* *Q8. Tìm các k đoạn đường đầu với lưu lượng truy cập cao trong giờ cao điểm cho tất cả các ngày làm việc.*

Bằng cách phân tích lịch sử di chuyển, các tuyến đường có lưu lượng cao có thể được phát hiện và mật độ giao thông có thể được điều chỉnh để cải thiện giao thông vận tải. Các giá trị lưu lượng của một đoạn đường được thiết lập như là số lượng các đối tượng chuyển động đi ngang qua trong giờ cao điểm ([7:00, 9:00] hoặc [16:00, 18:00]), nơi các phương thức vận chuyển được xem xét như: {ôtô, taxi, xe đạp, xe buýt}. Kỹ thuật tối ưu hóa được phát triển để cải thiện truy vấn một cách hiệu quả. Các phương thức vận tải tham gia của một đối tượng chuyển động là đại diện nhỏ gọn với mục đích cải thiện các chế độ kiểm tra sự tồn tại trong trường hợp các xác định được đưa ra trong một mô hình vận tải xảy ra thường xuyên. Một chỉ số được xây dựng trên các đơn vị chuyến đi để đẩy nhanh các thủ tục của việc truy cập dữ liệu. Bên cạnh đó, các truy vấn trên các đối tượng chuyển động, hệ thống hỗ trợ yêu cầu dữ liệu từ cơ sở hạ tầng. Mặc dù các đối tượng cơ sở hạ tầng nằm trong các môi trường khác nhau và được đại diện bởi các dữ liệu khác nhau, so sánh chúng trong cùng một bối cảnh vẫn còn có sẵn.

* *Q9. Với một tòa nhà, tìm thấy tất cả các điểm dừng xe buýt đó là trong vòng một bán kính 300 mét**.*

## Kết luận

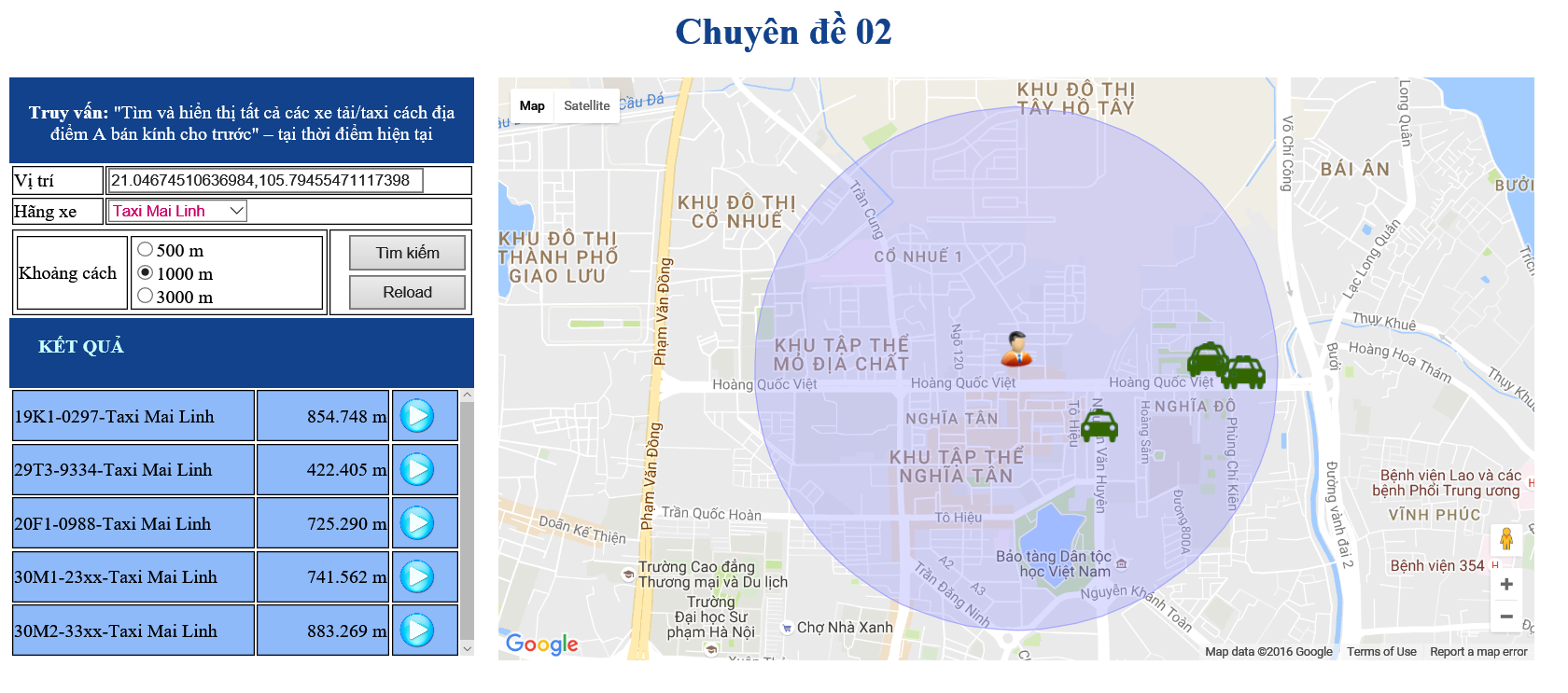
Chương 2 đã trình bày về kiến trúc hệ thống CSDL Secondo, trình bày về việc quản lý và truy vấn đối tượng chuyển động trong Secondo. Đồng thời còn trình bày về một thuật toán tối ưu hóa việc tìm đường đi ngắn nhất trong một đồ thị thứ tự vị từ.

# Chương 3. Thiết kế và thử nghiệm trên hệ thống lưu trữ dữ liệu chuyển động

Dựa trên những nghiên cứu được nêu ở chương 2, trong chương này sẽ trình bày về các mẫu truy vấn cơ sở dữ liệu chuyển động

## Truy vấn 1

|  |  |
| --- | --- |
| Mô tả | ***"Tìm và hiển thị tất cả các xe tải/taxi cách địa điểm A bán kính 500 m?" – tại thời điểm hiện tại*** |
| Input | - Tọa độ địa điểm A  - Tọa độ các xe đang hoạt động  - Khoảng cách cần tìm (bán kính) |
| Output | - Tọa độ địa điểm A  - Khoảng cách cần tìm (bán kính)  - Danh sách các xe |
| Truy vấn | **SELECT** \*  **FROM** ListVehicle  **WHERE** (distance > R) |



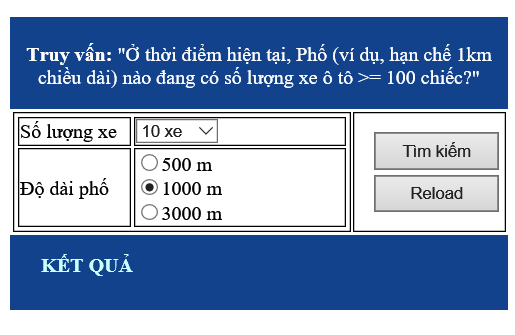
## Truy vấn 2

|  |  |
| --- | --- |
| Mô tả | ***Tìm tất cả các xe của hãng Taxi ABC đi qua đường Nguyễn Chí Thanh, Hà Nội từ lúc 11:00 – 12:00, vào ngày 2/8/2016?*** |
| Input | - Đường Nguyễn Chí Thanh  - Thời gian (bắt đầu – kết thúc): 11:00 – 12:00  - Ngày: 02/08/2016 |
| Output | - List danh sách các xe của hãng |
| Truy vấn | **SELECT** \*  **FROM** ListVehicleOfABC  **WHERE** (time > fromdate) and (time <= todate) |



## Truy vấn 3

|  |  |
| --- | --- |
| Mô tả | ***Ở thời điểm hiện tại, Phố (ví dụ, hạn chế 1km chiều dài) nào đang có số lượng xe ô tô >= 100 chiếc?*** |
| Input | - Số lượng xe ô tô: 100  - Hạn chế chiều dài: 1km |
| Output | - List danh sách phố thỏa mãn điều kiện truy vấn |
| Truy vấn | **SELECT** \*  **FROM** ListOfStreets  **WHERE** (max = 100) and (distance < 1000) |



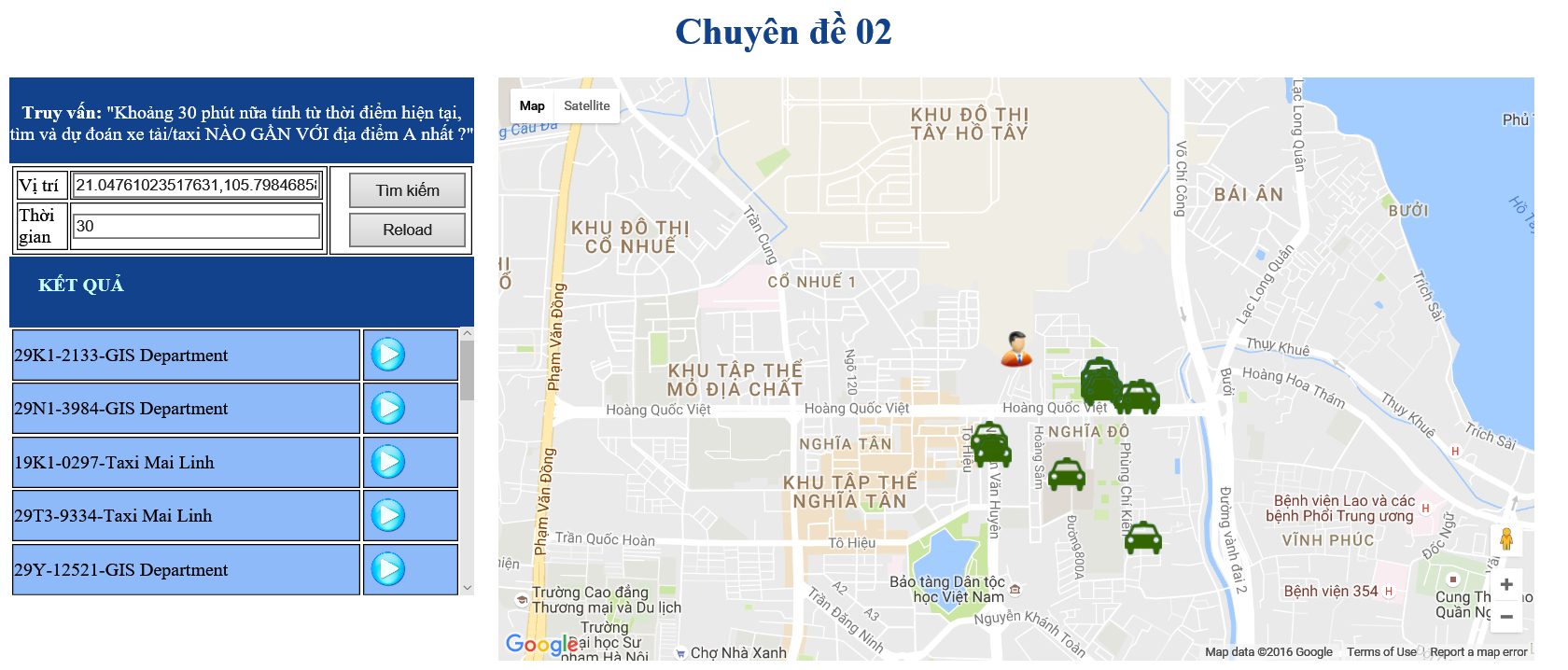
## Truy vấn 4

|  |  |
| --- | --- |
| Mô tả | ***"Khoảng 30 phút nữa tính từ thời điểm hiện tại, tìm và dự đoán tất cả các xe tải/taxi cách địa điểm A bán kính 500 m?"*** |
| Input | - Tọa độ điểm A  - Bán kính: 500m  - Thời gian dự đoán |
| Output | - Danh sách xe |
| Truy vấn | **SELECT** \*  **FROM** ListVehicle  **WHERE** (nexttime = 30) and (distance < 500) |



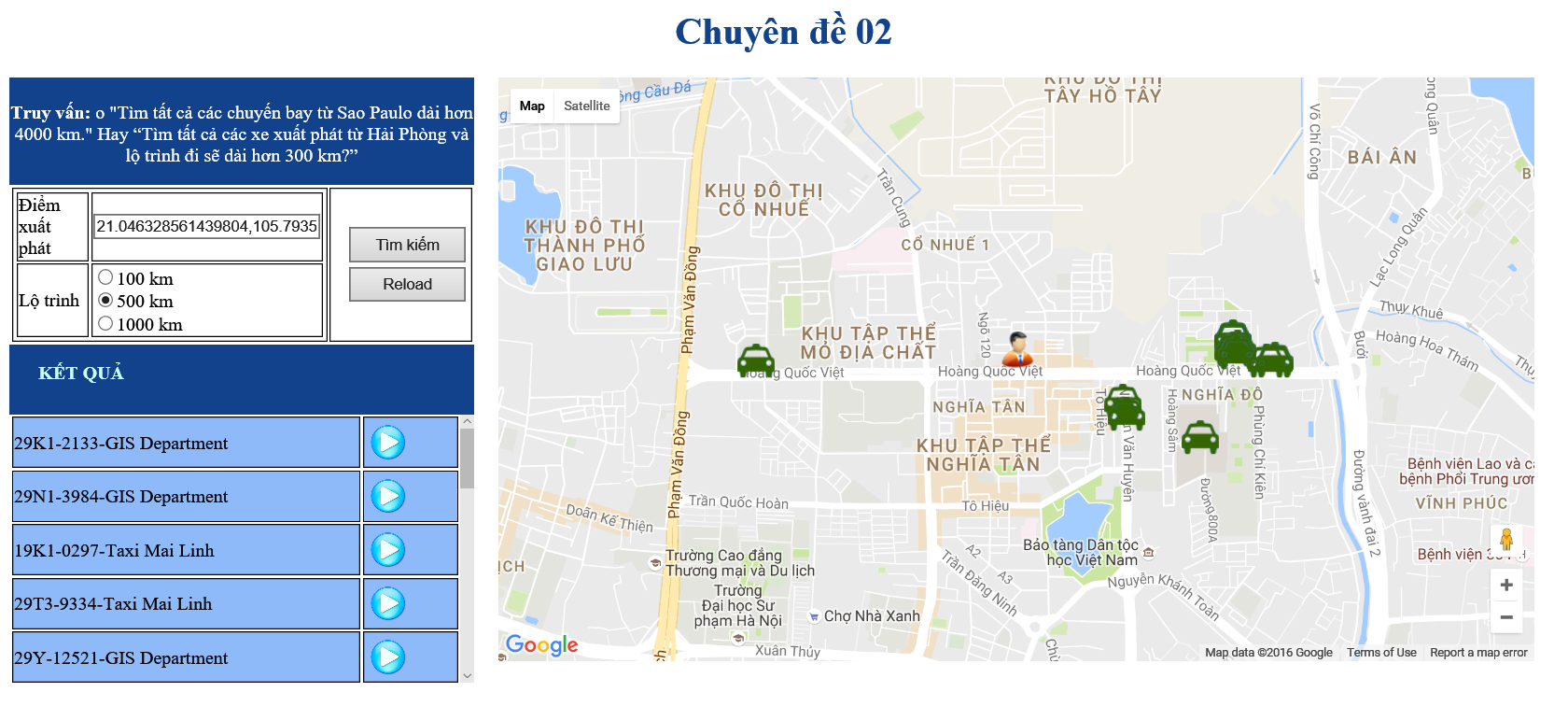
## Truy vấn 5

|  |  |
| --- | --- |
| Mô tả | ***"Khoảng 30 phút nữa tính từ thời điểm hiện tại, tìm và dự đoán xe tải/taxi NÀO GẦN VỚI địa điểm A nhất?":*** |
| Input | - Tọa độ điểm A  - Thời gian dự đoán  - Lộ trình trước đây |
| Output | - Danh sách xe |
| Truy vấn | **SELECT** \*  **FROM** ListVehicle  **WHERE** (nexttime = 30) |



## Truy vấn 6

|  |  |
| --- | --- |
| Mô tả | ***"Tìm tất cả các chuyến bay từ Sao Paulo dài hơn 4000 km." Hay “Tìm tất cả các xe xuất phát từ Hải Phòng và lộ trình đi sẽ dài hơn 300 km?”*** |
| Input | - Điểm xuất phát: Hải Phòng  - Lộ trình: 300km |
| Output | - Danh sách xe |
| Truy vấn | **SELECT** \*  **FROM** ListVehicle  **WHERE** (startPoint = “HaiPhong”) and (distance > 300000) |



# Kết luận

Nhóm nghiên cứu đã trình bày các kết quả thu được trong quá trình nghiên cứu cơ chế truy vấn các đối tượng chuyển động và xây dựng các mẫu truy vấn thực nghiệm trên cơ sở dữ liệu chuyển động thời gian thực.

Tuy vẫn còn nhiều vấn đề cần nghiên cứu như tối ưu hóa truy vấn dữ liệu không gian, thời gian nhưng các kết quả đạt được ban đầu cũng đáng được ghi nhận và là động lực để nhóm nghiên cứu có thể tìm hiểu những chuyên đề tiếp theo.