

**ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**LÊ VĂN TRUNG**

**NÂNG CẤP TÍNH NĂNG TÌM KIẾM CỦA  
PHẦN MỀM THƯƠNG MẠI ĐIỆN TỬ MAGENTO  
SỬ DỤNG SPHINX SEARCH**

Ngành: Công nghệ Thông tin

Chuyên ngành: Truyền dữ liệu và Mạng máy tính

Mã số:

**TÓM TẮT LUẬN VĂN THẠC SĨ  
NGÀNH CÔNG NGHỆ THÔNG TIN**

**Hà Nội - 2016**

# NÂNG CẤP TÍNH NĂNG TÌM KIẾM CỦA PHẦN MỀM THƯƠNG MẠI ĐIỆN TỬ MAGENTO SỬ DỤNG SPHINX SEARCH

*Đại học Công Nghệ - Đại học Quốc gia Hà Nội*

Luận văn thạc sĩ ngành: Công nghệ thông tin.

Mã số:

Người hướng dẫn khoa học: TS. Trần Trúc Mai

Học viên thực hiện luận văn: Lê Văn Trung

**Abstract:** Luận văn tìm hiểu về các kiến thức liên quan đến việc nâng cấp tính năng tìm kiếm của phần mềm thương mại điện tử Magento dựa trên một công cụ tìm kiếm toàn văn bản Sphinx Search. Dựa trên lý thuyết nghiên cứu về Magento và Sphinx Search xây dựng một chương trình tích hợp vào mã nguồn Magento để nâng cấp tính năng tìm kiếm mặc định có sẵn trong Magento. So sánh tính hiệu quả của việc nâng cấp tính năng tìm kiếm này dựa trên tiêu chí chất lượng tìm kiếm.

**Keyword:** Tìm kiếm trong Magento, Công cụ tìm kiếm Sphinx Search.

## **DANH SÁCH HÌNH VẼ**

|  |    |
|--|----|
| Hình 3.1. Tiến trình làm việc của Sphinx Search..... | 9  |
| Hình 4.1. Kết quả quá trình lập chỉ số.....          | 17 |
| Hình 5.1. Giao diện trang quản trị.....              | 19 |
| Hình 5.2. Giao diện trang tìm kiếm.....              | 19 |
| Hình 5.3. Giao diện trang kết quả tìm kiếm .....     | 20 |

## **DANH SÁCH BẢNG BIỂU**

|  |    |
|--|----|
| Bảng 3.1. Danh sách chức năng gộp kết quả tìm kiếm ..... | 15 |
| Bảng 4.1. Bảng Sphinx_catalogSearch_fulltext.....        | 17 |
| Bảng 5.1. Đánh giá kết quả tìm kiếm .....                | 21 |

## MỞ ĐẦU

### 1. ĐẶT VẤN ĐỀ

Ngày nay thương mại điện tử phát triển vô cùng mạnh mẽ bởi tốc độ sử dụng internet cùng với ngày càng nhiều các công nghệ hiện đại ra đời. Mọi người ngày càng ưa thích giao dịch dưới hình thức này bởi những thuận lợi của nó mang lại như tiện lợi, nhanh chóng, tiết kiệm chi phí, dễ dàng để tìm kiếm các mặt hàng cần mua, bán. Do đó yêu cầu đặt ra là xây dựng các gian hàng trực tuyến có khả năng hỗ trợ tìm kiếm hiệu quả và nâng cao để có thể thực hiện mua bán trực tuyến dễ dàng. Magento là phần mềm thương mại điện tử mã nguồn mở hoàn toàn miễn phí và cho phép người dùng thiết kế, cài đặt và chạy các gian hàng thương mại điện tử của riêng mình với sự linh hoạt cao. Tuy nhiên trong Magento chức năng tìm kiếm mặc định có nhiều hạn chế vì vậy ta cần nâng cấp tính năng tìm kiếm này để đáp ứng các yêu cầu ngày càng cao của khách hàng.

### 2. MỤC ĐÍCH CỦA LUẬN VĂN

Đề xuất sử dụng công cụ tìm kiếm toàn văn bản Sphinx Search để nâng cấp tính năng tìm kiếm mặc định trong Magento.

Xây dựng chương trình mở rộng tính năng tìm kiếm bằng cách tích hợp Sphinx Search vào mã nguồn Magento.

Đánh giá kết quả tìm kiếm trước và sau khi sử dụng Extension Sphinx Search.

### 3. BỐ CỤC CỦA LUẬN VĂN

Ngoài phần mở đầu, kết luận, danh mục, các đề tài có liên quan, luận văn gồm 05 chương được tổ chức như sau:

*Chương 1: Tổng quan về Magento và môi trường cài đặt:* Giới thiệu chung về hệ thống mã nguồn mở thương mại điện tử Magento và môi trường cài đặt Magento trong luận văn.

*Chương 2: Tìm kiếm trên Magento:* Trình bày về các công cụ tìm kiếm có sẵn trong Magento, những hạn chế của các loại tìm kiếm mặc định này và đặt ra bài toán cần nâng cấp tính năng tìm kiếm mặc định của hệ thống Magento.

*Chương 3: Sử dụng Sphinx Search nâng cấp tính năng tìm kiếm trong Magento:* Trình bày các đặc điểm của công cụ tìm kiếm sử dụng Sphinx Search đồng thời giải thích lý do sử dụng công cụ này.

*Chương 4: Triển khai nâng cấp tính năng tìm kiếm cho Magento bằng cách sử dụng Sphinx : Các bước triển khai xây dựng một extension mở rộng tính năng tính kiểm tích hợp vào Magento*

*Chương 5: Đánh giá kết quả: Tổng kết đánh giá kết quả việc nâng cấp tính năng tìm kiếm trên hệ thống mã nguồn mở Magento.*

Nội dung chi tiết từng chương sẽ được trình bày ở các phần sau.

# **Chương 1. TỔNG QUAN VỀ MAGENTO VÀ MÔI TRƯỜNG CÀI ĐẶT**

## **1.1. Tổng quan về Magento**

### **1.1.1. Giới thiệu chung**

Magento là phần mềm thương mại điện tử mã nguồn mở được sử dụng bởi các thương hiệu hàng đầu trên thế giới. Magento được cho là hệ thống thương mại điện tử linh hoạt và hoàn chỉnh nhất hiện nay với nhiều tính năng tiên tiến để thiết lập các gian hàng ảo, độc đáo, an toàn, thân thiện với các công cụ tìm kiếm, tạo điều kiện thuận lợi cho việc lập chỉ số trong công cụ tìm kiếm Google, giảm chi phí đầu tư vào quảng cáo. Magento cho phép người dùng tạo và quản lý các gian hàng trực tuyến một cách dễ dàng, nhanh chóng.

Hiện nay, không chỉ có Magento là giải pháp thương mại điện tử mã nguồn mở duy nhất trên thị trường, có những phần mềm mã nguồn mở khác như: Opencart, PrestaShop, osCommerce (more or less obsolete), Joomla với VirtureMart, Wordpress với e-commerce plugins like WooCommerce, Zen Cart.

Tuy nhiên Magento là nền tảng thương mại điện tử phổ biến nhất chiếm 25,6% đứng đầu thế giới thống kê năm 2014.

### **1.1.2. Lịch sử phát triển**

Bắt đầu từ năm 2001. Đến tháng 11/2015 phát hành phiên bản 2.0.

### **1.1.3. Tính năng của hệ thống mã nguồn mở Magento**

Magento là giải pháp nền tảng thương mại điện tử có tính năng phong phú, cung cấp các công cụ điều khiển các gian hàng trực tuyến linh hoạt. Magento tối ưu hóa công cụ tìm kiếm, quản lý danh mục sản phẩm và các khả năng marketing mạnh mẽ để tạo ra các website bán hàng tốt nhất phù hợp với yêu cầu của khách hàng.

## **1.2. Ưu điểm và nhược điểm của hệ thống mã nguồn mở**

Magento có nhiều ưu điểm nổi bật như: Mã nguồn mở sử dụng hoàn toàn miễn phí, thân thiện với người sử dụng, nhiều tính năng phong phú, thân thiện với các công cụ tìm kiếm... Bên cạnh những ưu điểm đó Magento vẫn còn tồn tại một số nhược điểm đó là: Chậm chạp, phức tạp đối với người dùng mới.

## **1.3. Môi trường cài đặt**

Website thương mại điện tử Magento trong đề tài nghiên cứu sử dụng phiên bản Magento Community Edition 1.9.1.0 được cài trên

LEMP Server (L-Linux, E-Nginx, M- Mysql, P-php) trong VPS(Virtual Private Server) của DigitalOcean.

### **1.3.1. Virtual Private Server của DigitalOcean.**

VPS - Virtual Private Server là phương pháp phân chia máy chủ vật lý thành nhiều máy chủ ảo.

DigitalOcean là một nhà cung cấp dịch vụ máy chủ ảo (VPS-virtual Private Server) của Mỹ có trụ sở tại thành phố New York.

### **1.3.2. LEMP Server**

LEMP server là một server chạy Linux (có thể là Ubuntu, CentOS...). Máy chủ web Nginx, cơ sở dữ liệu MySQL, Ngôn ngữ lập trình PHP (hoặc Perl/Python).

## **1.4. Kết luận chương 1**

Chương này tập trung giới thiệu 2 vấn đề chính. Vấn đề đầu tiên, giới thiệu tổng quan về Magento và những ưu điểm của mã nguồn mở Magento. Vấn đề tiếp theo, trình bày về môi trường cài đặt website Magento áp dụng trong luận văn.

## **Chương 2. TÌM KIẾM TRÊN MAGENTO**

### **2.1. Giới thiệu**

Trên Magento phiên bản hiện tại cung cấp công cụ tìm kiếm tiêu chuẩn với ba loại như sau:

- a) Tìm kiếm 'like'.
- b) Tìm kiếm toàn văn bản.
- c) Tìm kiếm kết hợp giữa like và tìm kiếm toàn văn bản.

### **2.2. Thuật toán nền tảng của các loại tìm kiếm trong Magento.**

#### **2.2.1. Tìm kiếm like**

Đây là phương thức tìm kiếm đơn giản nhất trong các phương thức tìm kiếm ở trên, áp dụng thuật toán tìm kiếm tuần tự để thực hiện việc tìm kiếm trong cơ sở dữ liệu. Trong Magento mỗi khi chúng ta nhập vào từ khóa cần tìm kiếm hệ thống sẽ ngắt chuỗi từ khóa thành các từ riêng và thực hiện tìm kiếm từng từ, sau đó kết hợp các kết quả tìm kiếm bằng biểu thức logic "or" sẽ được kết quả tìm kiếm cuối cùng.

#### **2.2.2. Tìm kiếm toàn văn bản**

Magento sử dụng tìm kiếm toàn văn bản trong MySQL để thực hiện tìm kiếm.

#### **2.2.3. Tìm kiếm kết hợp**

Kết hợp cả 2 thuật toán tìm kiếm trên.

Cả ba loại tìm kiếm này đều không đạt hiệu suất tìm kiếm tốt, thời gian tìm kiếm không đạt yêu cầu, kết quả tìm kiếm có sự nhiễu cao. Vì vậy cần đưa ra một giải pháp để nâng cấp tính năng tìm kiếm mặc định trong Magento.

### **2.3. Giải pháp nâng cấp tính năng tìm kiếm trên Magento.**

#### **2.3.1. Phương pháp tiếp cận**

Nâng cấp tính năng tìm kiếm là cải thiện về kết quả hiển thị và thời gian tìm kiếm. Để đáp ứng yêu cầu này thì ta quan tâm đến một số các lĩnh vực sau:

- Mức độ liên quan của từ khóa tìm kiếm và kết quả tìm kiếm.
- Khả năng sửa lỗi chính tả khi người dùng nhập vào từ khóa tìm kiếm.
- Khả năng tìm kiếm các từ đồng nghĩa.
- Tự động hoàn thành hoặc gợi ý cho các từ khóa tìm kiếm.
- Tăng tốc độ tìm kiếm.

#### **2.3.2. Các giải pháp mở rộng tính năng tìm kiếm cho Magento.**



Hiện tại có một số nền tảng để nâng cấp cũng như mở rộng tính năng tìm kiếm cho Magento như: Algolia Search, Solr, Sphinx, ElasticSearch, Amazon CloudSearch, Soocr Search, SLI learning Search.

### **2.3.3. Sử dụng Sphinx để nâng cấp tính năng tìm kiếm**

Để giải quyết vấn đề nâng cấp tính năng tìm kiếm. Giải pháp đưa ra là sử dụng công cụ tìm kiếm toàn văn bản Sphinx Search. Vì Sphinx Search có nhiều đặc điểm và tính năng phong phú ví dụ tốc độ lập chỉ số lớn, có nhiều hỗ trợ lựa chọn tìm kiếm.

## **2.4. Kết luận chương 2**

Sau khi cài đặt và cấu hình Magento trên LEMP server sử dụng VPS của Digital Ocean, Nghiên cứu sử dụng tính năng tìm kiếm cơ bản có sẵn trong Magento. Kết quả cho thấy:

- Tốc độ lập chỉ số (indexing) chậm so với các công cụ tìm kiếm khác.
- Tốc độ tìm kiếm cùng 1 từ khóa với lượng dữ liệu như nhau chậm.
- Hiệu suất tìm kiếm kém, các lựa chọn tìm kiếm rất hạn chế.

Với kết quả như vậy, Khi người sử dụng thực hiện chức năng tìm kiếm sản phẩm trên gian hàng được xây dựng bằng Magento sẽ trả lại những kết quả không mong muốn, và thời gian trả lại kết quả chậm.

Để giải quyết vấn đề này, Nghiên cứu cải thiện tính năng tìm kiếm của Magento bằng cách sử dụng Sphinx là nhu cấp thiết và phù hợp với thực tế. Sphinx đáp ứng được hầu hết các yêu cầu nâng cấp tìm kiếm, đồng thời là chương trình mã nguồn mở, rất phù hợp để nghiên cứu và phát triển.

Nội dung của Sphinx Search được giới thiệu trong chương 3.

## Chương 3. SỬ DỤNG SPHINX SEARCH NÂNG CẤP TÍNH NĂNG TÌM KIẾM TRÊN MAGENTO

### 3.1. Một số vấn đề về tìm kiếm toàn văn bản

Sphinx là công cụ tìm kiếm toàn văn bản, nên trước khi đi xa hơn chúng ta cần tìm hiểu một số vấn đề chung về tìm kiếm toàn văn bản và vì sao nó vượt trội hơn so với các tìm kiếm truyền thống.

#### 3.1.1. Khái niệm tìm kiếm toàn văn bản.

**Khái niệm:** Tìm kiếm toàn văn bản tên tiếng anh Full Text Search là một kỹ thuật được sử dụng để tìm kiếm một tài liệu được lưu trữ trên máy tính hoặc cơ sở dữ liệu. Trong FTS các công cụ tìm kiếm xem xét các tất cả các từ trong mỗi tài liệu được lưu trữ và cố gắng tìm ra những từ phù hợp yêu cầu của người sử dụng.

**Thuật toán đánh chỉ số:** Các thuật toán xây dựng chỉ số bao gồm 1 số loại như sau: Suffix tree , Inverted index , Citation index, Ngram index , Document-term matrix .

#### 3.1.2. Khái niệm chỉ số và thuật toán đánh chỉ số

**Khái niệm:** Một database index là một cấu trúc dữ liệu để cải thiện tốc độ tìm kiếm dữ liệu, Nếu không có index, các công cụ tìm kiếm sẽ quét tất cả các tài liệu điều đó sẽ tốn thời gian và khả năng tính toán.

### 3.2. Tổng quan về Sphinx Search

#### 3.2.1. Định nghĩa:

Sphinx Search: là một công cụ tìm kiếm toàn văn bản, cung cấp khả năng tìm kiếm nhanh chóng, độ chính xác cao, phù hợp với các ứng dụng của bên thứ ba. Nó có nhiều đặc điểm phù hợp và hỗ trợ cho việc tìm kiếm trên cơ sở dữ liệu SQL, tích hợp tốt vào các ngôn ngữ kịch bản như PHP, Python, Ruby, Perl và Java.

#### 3.2.2. Tính năng của Sphinx Search.

Sphinx Search có nhiều tính năng phong phú:

- Lập chỉ mục và tìm kiếm hiệu suất cao.
- Lập chỉ mục nâng cao và chứa công cụ truy vấn linh hoạt, tính năng phong phú tokenizer văn bản, ngôn ngữ truy vấn, có nhiều chế độ xếp hạng kết quả tìm kiếm khác nhau.
- Đạt được kết quả truy vấn hiệu quả sau khi sử dụng (SELECT với các biểu thức, WHERE, ORDER BY, GROUP BY, HAVING ...trên kết quả tìm kiếm văn bản).
- Có khả năng mở rộng đã được kiểm chứng lên đến hàng tỷ tài liệu, terabyte dữ liệu và hàng ngàn truy vấn mỗi giây.

- Dễ dàng kết hợp với nguồn dữ liệu SQL, XML và giao diện tìm kiếm SphinxQL, SphinxAPI, hoặc SphinxSE.
- Dễ dàng mở rộng quy mô với các tìm kiếm phân tán.
- Có tốc độ lập chỉ mục cao (tối đa 10-15 MB/giây cho mỗi core trên một tiêu chuẩn nội bộ).
- Có tốc độ tìm kiếm cao (tối đa 150-250 truy vấn/ giây cho mỗi core tìm kiếm trên 1.000.000 tài liệu, 1,2 GB dữ liệu trong tiêu chuẩn nội bộ).
- Có khả năng mở rộng cao (Chỉ số cụm được biết đến nhiều nhất qua 3.000.000.000 tài liệu, đỉnh cao là 50.000.000 truy vấn/ngày).
- Cung cấp khả năng xếp hạng mức độ phù hợp của tài liệu tốt bằng cách kết hợp xếp hạng cụm từ gần và xếp hạng thống kê (BM25).
- Hỗ trợ nhiều trường full-text cho mỗi tài liệu (mặc định lên tới 32 trường)
- Hỗ trợ nhiều thuộc tính bổ sung cho mỗi tài liệu (ví dụ group, timestamps, etc...)
- Hỗ trợ stopwords.
- Hỗ trợ morphological word từ các từ điển.
- Hỗ trợ loại trừ tokenizing.
- Hỗ trợ UTF-8 encoding.

### **3.2.3. Lịch sử phát triển.**

Bắt đầu phát triển từ năm 2001 bởi Andrew Aksyonoff (Một nhà phát triển website của nga).

### **3.2.4. Cài đặt Sphinx trong môi trường Linux**

### **3.2.5. Cách sử dụng Sphinx Search.**

*Bước 1: Tạo file cấu hình*

*Bước 2: Chạy indexer để tạo full-text index từ dữ liệu của bạn.*

*Bước 3: Truy vấn các chỉ số vừa mới được tạo ra.*

## **3.3. Cơ chế hoạt động của Sphinx Search**

### **3.3.1. Các thành phần trong Sphinx**

Hiện tại gói đầy đủ Sphinx Search bao gồm các phần mềm sau đây:

**Indexer** : là một tiện ích tạo ra fulltext indexes.

**Searchd**: là một daemon cho phép phần mềm bên ngoài (ví dụ các ứng dụng web) có thể tìm kiếm thông qua fulltext indexes (Chỉ số toàn văn bản).

**SphinxAPI:** là một tập hợp các thư viện client API cho các ngôn ngữ kịch bản web phổ biến (PHP, Python, Perl, Ruby).

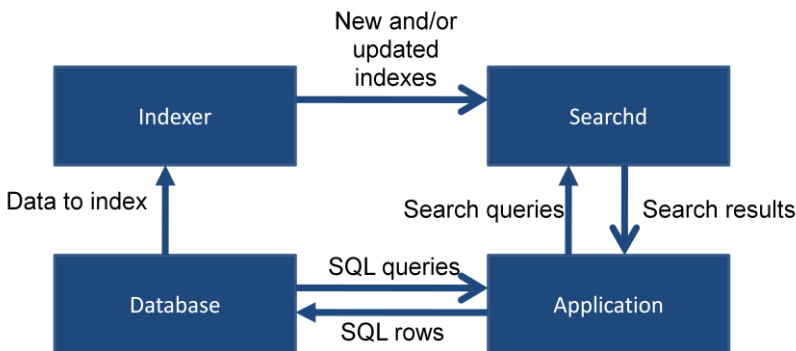
**Spelldump:** Một công cụ dòng lệnh (command-line) để trích xuất các items từ một ispell hoặc MySpell định dạng từ điển để giúp tùy chỉnh chỉ số(index) của bạn.

**Indextool:** Là một tiện ích sửa lỗi về các chỉ số (index) được thêm vào trong phiên bản 0.9.9-RC2.

**Wordbreaker:** một tiện ích để phá vỡ các từ ghép thành các từ đơn riêng biệt, được thêm vào trong phiên bản 2.1.1

### 3.3.2. Tiến trình làm việc.

Một giải pháp tìm kiếm đầy đủ bao gồm bốn thành phần chính sau:



Hình 3.1. Tiến trình làm việc của Sphinx Search

#### Trình duyệt website máy khách:

Người dùng nhập một chuỗi tìm kiếm vào chương trình máy khách( ví dụ như trình duyệt web) Sau đó chương trình sẽ gửi một truy vấn tìm kiếm đến Searchd, sau đó hiển thị các kết quả tìm kiếm.

#### Data source (Nguồn dữ liệu).

Đây là nơi lưu trữ dữ liệu của bạn và được truy vấn bởi các chương trình indexer. Nguồn dữ liệu được sử dụng phổ biến là MySQL. Hoặc một SQL server để lưu trữ. Nhưng đây không phải yêu cầu bắt buộc. Nó có thể làm việc tốt với các nguồn dữ liệu không phải là SQL.

#### Indexer:

Chương trình này tìm nạp dữ liệu từ các nguồn dữ liệu (Data source) và tạo ra một full-text index của dữ liệu đó. Bạn sẽ cần phải chạy indexer định kỳ, tùy thuộc vào yêu cầu cụ thể của bạn.

### Searchd:

Đây là một chương trình nói chuyện với your client program (ví dụ trình duyệt website của bạn), và sử dụng các full-text được xây dựng bởi indexer để nhanh chóng xử lý các truy vấn tìm kiếm. Tuy nhiên Searchd thực hiện nhiều việc hơn là chỉ tìm kiếm ví dụ như tách một chuỗi thành các từ khóa (hay còn gọi là tokenizing) hoặc xây dựng snippets(đoạn ngắn) và một số nhiệm vụ khác.

### 3.3.3. Lập chỉ số

Lập chỉ số được thực hiện bằng việc chạy chương trình indexer. Khi indexer bắt đầu khởi chạy, một kết nối với cơ sở dữ liệu được thiết lập bằng cách sử dụng các thông tin cung cấp trong tập tin cấu hình. Sau đó truy vấn chính sql\_query được sử dụng để xác định các trường dữ liệu sẽ lập chỉ mục. Một khi việc xác định các trường cần lập chỉ mục được hoàn thành tiến hành đóng kết nối tới cơ sở dữ liệu và chuyển sang giai đoạn sắp xếp, phân loại index.

Kết nối tới nguồn dữ liệu sẽ được lập chỉ mục sẽ được xác định như sau trong file cấu hình:

```
source src1 {
    type          = mysql
    sql_host      = localhost
    sql_user      = test
    sql_pass      =
    sql_db        = test
    sql_port      = 3306 # optional, default is
3306
}
```

### Truy vấn chính xác định các trường sẽ được lập chỉ số

```
sql_query      = \
SELECT         id,          title,          content,
UNIX_TIMESTAMP(publish_date) \
AS publish_date, author_id FROM posts
```

**Một số các tùy chọn trong lập chỉ mục đó là xác định stop words và shorts words.**

**Stop words:** là những từ khóa thường xuyên xảy ra, bạn có thể bỏ qua những từ này khi lập chỉ mục và tìm kiếm. Ví dụ “i, the, and, to, a, it...”). Loại bỏ stop word có thể cải thiện đáng kể thời gian lập chỉ mục và kích thước chỉ mục.

Để xử lý stop word chúng ta tiến hành tạo file cấu hình Sphinx.conf như sau:

```
index test1
{
path = /var/lib/Sphinx/data/test1
source = src1
stopwords = /var/lib/Sphinx/stopwords.txt
}
```

Tập tin stopwords.txt là tập tin văn bản bao gồm các từ khóa được bỏ qua khi làm việc với index test1.

**Short words:** Là lựa chọn xác định chiều dài từ khóa tối thiểu được lập chỉ mục. Từ khóa ngắn hơn giới hạn này sẽ không được lập chỉ mục.

Sử dụng min\_word\_len để xác như sau:

```
index test1
{
path = /var/lib/Sphinx/data/test1
source = src1
min_word_len = 3
}
```

Ví dụ từ khóa “i” và “am” sẽ không được lập chỉ mục. Từ khóa “you” sẽ được lập chỉ mục vì có độ dài

### **3.3.4. Tìm kiếm.**

Chương trình Searchd được sử dụng để tìm kiếm trên chỉ số được tạo ra. Có 3 cách truy cập vào Searchd đó là: Qua API, SphinxSE và SphinxQL.

Hay nói cách khác bạn có thể gửi các tìm kiếm thông qua API hay qua các truy vấn của MySQL.

Triển khai API cho Sphinx

Sphinx đi kèm với một số Searchd API, Tại thời điểm hiện tại Sphinx API được triển khai cho ngôn ngữ PHP, Python, Java.

### Tìm kiếm sử dụng client API

Các bước để thực hiện tìm kiếm sử dụng API. Sử dụng các chỉ số và cấu hình đã tạo ra ở phần trước, bổ sung cấu hình liên quan đến Searchd và tạo một file PHP để tìm kiếm các chỉ số sử dụng Sphinx thực hiện client API cho PHP.

Các bước triển khai như sau:

#### (1) Thêm cấu hình Searchd vào file cấu hình:

```
Searchd {
listen          = 9312
log              =
/usr/local/Sphinx/var/log/Searchd.log
query_log       =
/usr/local/Sphinx/var/log/query.log
max_children    = 30
pid_file        =
/usr/local/Sphinx/var/log/Searchd.pid
}
```

Giải thích các tùy chọn:

Listen: Lựa chọn này xác định địa chỉ ip và cổng được Searchd sử dụng.

Log: Nơi chứa file log chứa tất cả các sự kiện của Searchd.

Query\_log: Tên tập tin chứa các lịch sử truy vấn tìm kiếm.

max\_children: Xác định số lượng tối đa các tìm kiếm chạy đồng thời. Mặc định là 0 (không giới hạn)

pid\_file: Tên file của quá trình Searchd ID.

#### (2) Bắt đầu chạy Searchd.

```
/usr/bin/Searchd -c /etc/Sphinx/Sphinx.conf
```

(3) Sao chép tập tin Sphinxapi.php vào thư mục đang làm việc.

```
mkdir /path/to/your/webroot/Sphinx
```

```
cd /path/to/your/webroot/Sphinx
```

```
cp /path/to/Sphinx-0.9.9/api/Sphinxapi.php ./
```

**(4) Tạo một kịch bản simple\_Search.php để sử dụng lớp PHP client API thực hiện tìm kiếm trên chỉ số đã được tạo ra trong phần trước.**

```
<?php require_once('Sphinxapi.php');
// Instantiate the Sphinx client
$client = new SphinxClient();
// Set Search options
$client->SetServer('localhost', 9312);
$client->SetConnectTimeout(1);
$client->SetArrayResult(true);
// Query the index
$results = $client->Query('php');
// Output the matched results in raw format
print_r($results['matches']);
```

Các phương thức được sử dụng :

SetServer(\$host, \$port). Đây là phương thức xác định tên máy chủ và cổng giao tiếp của Searchd. Mặc định host='localhost', và port='9312'.

SetConnectTimeout(\$timeout). Xác định thời gian tối đa để kết nối tới máy chủ.

SetArrayResult(\$arrayresult).Xác định kết quả tìm kiếm sẽ trả về dưới dạng một mảng hoặc một hàm băm (hash). Nếu (\$arrayresult là true thì kết quả tìm kiếm sẽ được trả lại giá trị dưới dạng một mảng.

\$client->Query('php'): Thực tìm kiếm từ khóa 'php'. Nó sẽ trả về một mảng các kết quả phù hợp.

Trong Sphinx cung cấp một số chế độ tìm kiếm để xác định các kết quả phù hợp với các yêu cầu của người sử dụng. Chúng được gọi là Matching Modes (chế độ phù hợp) được xác định bằng cách sử dụng phương thức setMatchMode().

### **3.3.5. Matching Modes (Chế độ phù hợp)**

Khi các chỉ số toàn văn bản được tạo ra ở phần trên. Để thực hiện việc tìm kiếm trên các chỉ số này, Sphinx có các chế độ tìm kiếm khác nhau được gọi là chế độ phù hợp để tìm kiếm kết quả.

Các chế độ tìm kiếm có sẵn trong Sphinx là:



SPH\_MATCH\_ALL: Kết quả trả về phù hợp với tất cả các từ khóa tìm kiếm.

SPH\_MATCH\_ANY: Kết quả trả về phù hợp với bất kỳ từ khóa tìm kiếm nào.

SPH\_MATCH\_PHRASE: Kết quả trả về phải phù hợp với cả cụm từ tìm kiếm.

SPH\_MATCH\_BOOLEAN: Chọn chế độ tìm kiếm được xây dựng bằng biểu thức boolean.

SPH\_MATCH\_EXTENDED: Chế độ tìm kiếm được xây dựng bằng các biểu thức trong ngôn ngữ riêng của Sphinx.

SPH\_MATCH\_EXTENDED2: Chế độ tìm kiếm mở rộng từ chế độ phù hợp từ SPH\_MATCH\_EXTENDED.

SPH\_MATCH\_FULLSCAN : Ở chế độ tìm kiếm này, tất cả các dữ liệu được lập chỉ số sẽ được trả về, áp dụng trong trường hợp từ khóa tìm kiếm là rỗng.

### **3.3.6. Xếp hạng kết quả tìm kiếm.**

Xếp hạng trong Sphinx là quá trình cấu hình. Nó kèm theo một khái niệm gọi là ranker. Một Ranker có thể được định nghĩa là một hàm có đầu vào là document và query và đầu ra là một giá trị liên quan. Sphinx sử dụng một thuật toán cụ thể để xác định ranker.

Các kết quả xếp hạng phụ thuộc vào chế độ phù hợp đã nêu ở trên.

### **3.3.7. Các Rankers được xây dựng sẵn**

**Trong Sphinx xây dựng sẵn một số ranker như sau:**

SPH\_RANK\_PROXIMITY\_BM25.

SPH\_RANK\_BM25.

SPH\_RANK\_NONE.

SPH\_RANK\_WORDCOUNT.

SPH\_RANK\_PROXIMITY.

SPH\_RANK\_MATCHANY.

SPH\_RANK\_FIELDMASK.

SPH\_RANK\_SPH04.

SPH\_RANK\_EXPR.

### 3.3.8. Sắp xếp kết quả tìm kiếm

Mặc định kết quả tìm kiếm được sắp xếp theo mức độ liên quan tuy nhiên trong một số trường hợp người sử dụng muốn sắp xếp kết quả tìm kiếm theo ý, vì vậy Sphinx có hỗ trợ một số chế độ sắp xếp đó là:

SPH\_SORT\_ATTR\_DESC: Sắp xếp theo thứ tự giảm dần của thuộc tính (thuộc tính có giá trị lớn nằm đầu tiên);

SPH\_SORT\_ATTR\_ASC: Sắp xếp theo thứ tự tăng dần của thuộc tính (thuộc tính có giá trị nhỏ nằm đầu tiên)

SPH\_SORT\_TIME\_SEGMENTS: Sắp xếp theo thứ tự thời gian.

Và một số chế độ sắp xếp khác.

### 3.3.9. Nhóm kết quả tìm kiếm

Sphinx cung cấp các chế độ group khác nhau. Chúng được kích hoạt bằng cách gọi phương thức SetGroupBy() trong API. Các chức năng được xây dựng sẵn trong Sphinx là:

| Phương thức           | Ý nghĩa:                          |
|-----------------------|-----------------------------------|
| SPH_GROUPBY_DAY       | Nhóm theo ngày                    |
| SPH_GROUPBY_WEEK      | Nhóm theo tuần                    |
| SPH_GROUPBY_MONT<br>H | Nhóm theo tháng                   |
| SPH_GROUPBY_YEAR      | Nhóm theo năm                     |
| SPH_GROUPBY_ATTR      | Sử dụng giá trị attribute để nhóm |

*Bảng 3.1. Danh sách chức năng gộp kết quả tìm kiếm*

## 3.4. Kết luận chương 3

Nội dung trong chương trình bày đặc điểm của công cụ tìm kiếm toàn văn bản Sphinx Search, các tính năng chính và cơ chế hoạt động.

Nội dung của chương tiếp theo sẽ trình bày việc triển khai tích hợp Sphinx Search vào Magento để nâng cấp tính năng tìm kiếm của Magento.

## **Chương 4. TRIỂN KHAI NÂNG CẤP TÍNH NĂNG TÌM KIẾM CHO MAGENTO BẰNG CÁCH SỬ DỤNG SPHINX**

### **4.1. Yêu cầu**

Bài toán nâng cấp tính năng tìm kiếm trên Magento sử dụng công cụ Sphinx được giải quyết bằng cách xây dựng phần mở rộng theo chuẩn của Magento để tích hợp vào mã nguồn. Phần mở rộng này sẽ thay thế tìm kiếm mặc định của Magento tạo ra một tìm kiếm thân thiện và mạnh mẽ. Đáp ứng một số yêu cầu cụ thể sau:

- Trả về kết quả tìm kiếm nhanh chóng.
- Trả về các kết quả liên quan phù hợp nhất.

Khi một người dùng thực hiện tìm kiếm trong gian hàng trực tuyến của bạn, Extention Magento Sphinx Search sẽ truy vấn tìm kiếm trong Sphinx để trả về kết quả cho người dùng.

### **4.2. Phân tích thiết kế**

#### **4.2.1. Chức năng lập chỉ số từ bảng dữ liệu của Magento**

Trong cơ sở dữ liệu của Magento xây dựng một bảng dữ liệu mới. Đặt tên bảng là *Sphinx\_catalogSearch\_fulltext*, Bảng này chứa các trường dữ liệu cần lập chỉ số.

Cài đặt công cụ Sphinx Search trên máy chủ và cấu hình trong file *Sphinx.conf*.

Chạy chức năng index trong Sphinx để tạo chỉ số từ bảng *dữ liệu vừa tạo ra ở trên*.

#### **4.2.2. Chức năng lựa chọn các chế độ tìm kiếm**

Sau khi các chỉ số được tạo ra bằng Sphinx ta tiến hành lựa chọn các chế độ tìm kiếm phù hợp với yêu cầu bao gồm:

- Chọn chế độ tìm kiếm
- Đánh trọng số các trường để ưu tiên hiển thị kết quả.
- Chọn chế độ xếp hạng tìm kiếm.
- Chọn chế độ liên quan của các từ khóa tìm kiếm.

### **4.3. Triển khai xây dựng**

#### **4.3.1. Tạo bảng chứa dữ liệu các trường sẽ được lập chỉ mục**

Xây dựng bảng *Sphinx\_catalogSearch\_fulltext* để lưu các dữ liệu cần index:

| Tên             | Loại dữ liệu |
|-----------------|--------------|
| product_id      | Int(10)      |
| Store_id        | Smallint(5)  |
| Name            | Varchar(255) |
| Name_attributes | longtext     |
| Category        | Varchar(255) |
| Data_index      | longtext     |

*Bảng 4.1. Bảng Sphinx\_catalogSearch\_fulltext*

### 4.3.2. Cài đặt Sphinx trên Server và cấu hình

### 4.3.3. Lập chỉ số dữ liệu

#### Chạy chương trình index để lập chỉ số:

```
[root@Tuan4 ~]# /usr/bin/indexer --all
Sphinx 2.0.9-id64-release (rel20-r4115)
Copyright (c) 2001-2013, Andrew Aksyonoff
Copyright (c) 2008-2013, Sphinx Technologies Inc (http://sphinxsearch.com)

using config file '/etc/sphinx/sphinx.conf'...
indexing index 'fulltext'...
WARNING: Attribute count is 0: switching to none docinfo
collected 270 docs, 0.4 MB
sorted 0.1 Mhits, 100.0% done
total 270 docs, 401817 bytes
total 0.064 sec, 6193710 bytes/sec, 4161.84 docs/sec
total 2 reads, 0.000 sec, 30.2 kb/call avg, 0.0 msec/call avg
total 6 writes, 0.000 sec, 21.8 kb/call avg, 0.0 msec/call avg
```

*Hình 4.1. Kết quả quá trình lập chỉ số*

### 4.3.4. Lựa chọn các chế độ tìm kiếm

Đoạn code chương trình lựa chọn chế độ tìm kiếm.

```
public function getSphinxAdapter() {
    require_once(Mage::getBaseDir('lib') .
DIRECTORY_SEPARATOR . 'Sphinxapi.php');

    $Sphinx = new SphinxClient();

    $host =
Mage::getStoreConfig('SphinxSearch/server/host');
```

```

        $port = Mage::getStoreConfig('SphinxSearch/server/port');
        if (empty($host)) {
            return $Sphinx;
        }
        if (empty($port)) {
            $port = 9312;
        }
        $Sphinx->SetServer($host, $port);
        $Sphinx->SetMatchMode(SPH_MATCH_EXTENDED2);
        $Sphinx->setFieldWeights(array(
            'name' => 7,
            'category' => 1,
            'name_attributes' => 1,
            'data_index' => 3
        ));
        $Sphinx->setLimits(0, 200, 1000, 5000);

        $Sphinx->SetRankingMode(SPH_RANK_SPH04, "");

        return $Sphinx;
    }

```

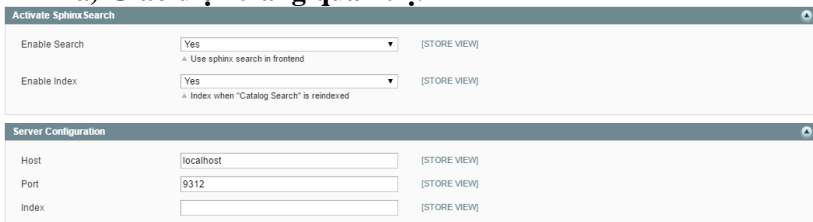
#### 4.4. Kết luận chương 4

Chương này của luận văn đã sử dụng các phần nội dung lý thuyết tìm hiểu ở phần trên để xây dựng một chương trình tích hợp vào Magento để nâng cấp tính năng tìm kiếm mặc định sẵn có dựa trên nền tảng công cụ tìm kiếm Sphinx Search đáp ứng các yêu cầu tìm kiếm của khách hàng trên các website thương mại điện tử sử dụng Magento.

## Chương 5. ĐÁNH GIÁ KẾT QUẢ

### 5.1. Kết quả xây dựng

#### a) Giao diện trang quản trị:



The screenshot displays two configuration panels in the Magento Admin interface. The first panel, titled "Activate Sphinx Search", contains two rows: "Enable Search" and "Enable Index". Both have a dropdown menu set to "Yes" and a "[STORE VIEW]" link. The second panel, titled "Server Configuration", contains three rows: "Host" (localhost), "Port" (9312), and "Index" (empty), each with a "[STORE VIEW]" link.

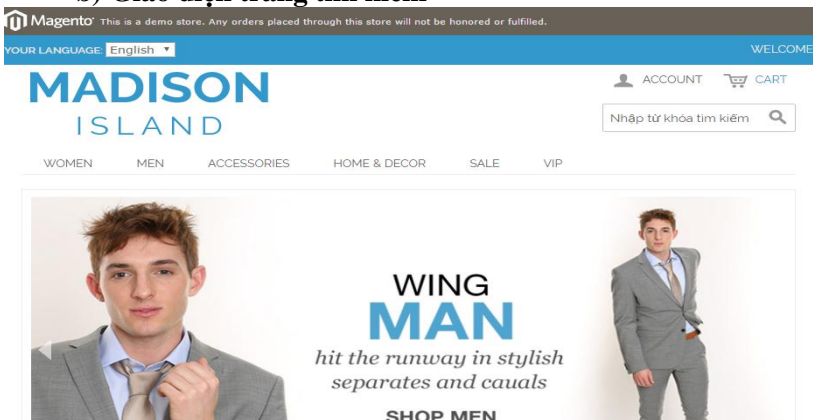
| Activate Sphinx Search |  |
|------------------------|--|
| Enable Search          | Yes [STORE VIEW]<br>▲ Use sphinx search in frontend            |
| Enable Index           | Yes [STORE VIEW]<br>▲ Index when "Catalog Search" is reindexed |

| Server Configuration |                        |
|----------------------|------------------------|
| Host                 | localhost [STORE VIEW] |
| Port                 | 9312 [STORE VIEW]      |
| Index                | [STORE VIEW]           |

Hình 5.1. Giao diện trang quản trị.

#### b) Giao diện trang tìm kiếm



Hình 5.2. Giao diện trang tìm kiếm.

#### c) Giao diện trang kết quả tìm kiếm

## SEARCH RESULTS FOR 'PLAID COTTON SHIRT'



Hình 5.3. Giao diện trang kết quả tìm kiếm

### 5.2. Đánh giá về kết quả xây dựng.

Chất lượng kết quả tìm kiếm bằng sphinx trả về tốt hơn tìm kiếm mặc định trong Magento.

| Từ khóa tìm kiếm | Sphinx Search  | Search default in Magento   |
|------------------|--|---|
| ANN ANKLE BOOT   | <p>Trả về 01 mặt hàng phù hợp.</p> <p>1.ANN ANKLE BOOT</p> | <p>Trả về 6 mặt hàng</p> <p>1.ANN ANKLE BOOT</p> <p>2.THE ESSENTIAL BOOT CUT JEAN.</p> <p>3.FLAT FRONT TROUSER.</p> <p>4.BARCLAY D'ORSAY PUMP, NUDE.</p> <p>5.A TALE OF TWO CITIES.</p> <p>6.FIRE [KALIMA REMIX] BY UNANNOUNCED GUEST</p> |

*Bảng 5.1. Đánh giá kết quả tìm kiếm*

**5.3. Kết luận chương 5.**

Nội dung của chương thể hiện sự so sánh các kết quả tìm kiếm giữa tìm kiếm mặc định trong Magento và tìm kiếm sử dụng Sphinx qua đó thấy được hiệu quả của việc nâng cấp tính năng tìm kiếm của phần mềm thương mại điện tử Magento.



## KẾT LUẬN

Tìm kiếm trên một gian hàng trực tuyến bằng Magento là yêu cầu cấp bách của người sử dụng. Tuy nhiên chức năng tìm kiếm có sẵn trong Magento có nhiều hạn chế về chất lượng và thời gian tìm kiếm chính vì vậy đề tài luận văn “Nâng cấp tính năng tìm kiếm của hệ thống mã nguồn mở Magento sử dụng Sphinx” đã giải quyết yêu cầu thực tế trên, luận văn tập trung tìm hiểu, nghiên cứu và đạt được một số kết quả sau:

### **Nghiên cứu tài liệu để trình bày các vấn đề sau:**

- Trình bày kiến thức cơ bản về hệ thống mã nguồn mở Magento, các chức năng chính, ưu điểm và nhược điểm của hệ thống.

- Tìm hiểu về chức năng tìm kiếm trên hệ thống mã nguồn mở Magento, hạn chế của chức năng tìm kiếm mặc định và các giải pháp nâng cấp tính năng tìm kiếm mặc định này.

- Nghiên cứu các đặc điểm của công cụ tìm kiếm toàn văn bản Sphinx: Các chức năng chính, cơ chế làm việc, cách tích hợp vào Magento.

- Nghiên cứu cách tích hợp Sphinx Search vào Magento để nâng cấp tính năng tìm kiếm.

Hướng phát triển tiếp theo của đề tài sẽ là nghiên cứu các hướng sau:

Tối ưu hóa chương trình đã xây dựng , cải thiện tốc độ và chức năng lập chỉ số, cải thiện kết quả tìm kiếm về thời gian và hiệu quả.

# TÀI LIỆU THAM KHẢO

## 1. Tài liệu Tiếng Việt

[1] Nguyễn Kim Quy (2006), “*Một số vấn đề liên quan đến SEARCH ENGINE, ứng dụng trong TEXT*”, Luận văn Thạc sỹ công nghệ thông tin, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội, tr. 42-57.

[2] Nguyễn Thị Hoàng Yến (2015), “*Cải thiện tốc độ hoạt động của Website thương mại điện tử sử dụng mã nguồn mở Magento thông qua kỹ thuật VARNISH*”, Luận văn Thạc sỹ công nghệ thông tin, Trường Đại học Công Nghệ, Đại học Quốc gia Hà Nội, tr. 1-23.

## 2. Tài liệu tiếng Anh

[3] Abbas Ali (2011), “Sphinx Search Beginner’s Guide”, Packt Publishing, Birmingham, B27 6PA, UK.

[4] Andrew Aksyonoff (2011), “Introduction to Search with Sphinx”, O’Reilly, United States of America.

## 3. Tài liệu tiếng Anh

[5] <http://www.vijaywebsolutions.com/advantages-disadvantages-of-Magento-ecommerce-platform/>

[6] <http://www.shoppimon.com/content/2015/01/optimizing-Magento-performance-solutions-help-speed-shop/>

[7] [https://en.wikipedia.org/wiki/Apache\\_Solr](https://en.wikipedia.org/wiki/Apache_Solr)

[8] <https://www.elastic.co>

[9] <http://www.sooqr.com/>.

[10] [https://en.wikipedia.org/wiki/Full\\_text\\_Search](https://en.wikipedia.org/wiki/Full_text_Search)

[11] <https://www.mysql.com/>

[12] [https://en.wikipedia.org/wiki/Suffix\\_tree](https://en.wikipedia.org/wiki/Suffix_tree)

[13] Digital Ocean. <http://www.digitalocean.com/>

[14] <http://Magento.com>

[15] <http://www.Magentoecommerce.com/knowledge-base/entry/Magento-for-dev-part-1-introduction-to-Magento.>

[16] <http://blog.aheadworks.com/2014/10/e-commerce-platforms-share-investigation-october-2014/>