EasyFunc Retrieval

# Apache Lucene

## Giới thiệu

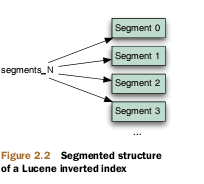
### Sơ lược

### Tổng quan

## Các thuật ngữ

### Indexing

* IndexWriter: Nơi xử lý index tài liệu
* Directory: Nơi lưu trữ tài liệu sau khi được index
* Document: Tài liệu dùng để đánh index
* Field: Đơn vị của tài liệu
* Token: Một từ hay cụm từ được analyzer trước khi được index
* Inverted index:



Với cách cấu trúc index thông thường, chúng ta sẽ quản lý các term dựa vào tài liệu. Ví dụ như ta sẽ nói tài liệu A có nội dụng gồm 3 term: “football”, “sport”, “player”.Như vậy, khi có yêu cầu tìm kiếm về term “football” chúng ta phải quét toàn bộ trang tài liệu để tìm vị trí của term “football”. Điều này sẽ khiến cho việc tìm kiếm của chúng ta không thực sự tốt.

Inverted index là cách mà chúng ta tạo ra một bảng vị trí của các term –tương tự như phần mục lục của một cuốn sách. Theo đó, chúng ta sẽ lập ra một bảng các term đi kèm với vị trí xuât hiện của nó trên toàn bộ tập tài liệu. Việc này có một số lợi ích như sau:

* Do độ dài của 1 term ngắn hơn độ dài của một tài liệu nên việc tìm kiếm trên toàn bộ term sẽ nhanh hơn tìm kiếm trên toàn bộ tài liệu
* Mỗi term luôn có một định danh duy nhất cho nên chúng ta có được kết quả tìm kiếm ngay trong lần hit đâu tiên( Best case: 1, Worst case: n). Với tài liệu chúng ta luôn phải tìm kiếm toàn bộ
* Field Option
  + Index: Nội dung của field được dùng trong việc tìm kiếm. Nếu không được index, nội dụng của field đấy sẽ không được sử dụng trong khi tìm kiếm
  + Stored: Nội dung của field sẽ giữ nguyên(Không bị analyzer) và lưu xuống bộ nhớ
* TermVector: Giống như một inverted index cho document. Với thuộc tính này chúng ta có thể dễ dàng truy xuất tất cả các các term trong một tài liệu. Rất có ich

### Searching