

Rails Full Text Search with Ferret

Jens Krämer

<http://www.jkraemer.net/>

<http://www.webit.de/>



```
$ gem install ferret
```

```
http://ferret.davebalmain.com/
```

The Problem



Index

Documents

Fields

Indexing

```
index = Ferret::Index.new :path => '/tmp/index'

books.each do |book|
  index << { :content => book.content,
            :title    => book.title,
            :author   => book.author }
end

index.optimize
index.close
```

Searching

```
index = Ferret::Index.new :path => '/tmp/index'

index.search_each('ruby') do |doc_id, score|
  doc = index[doc_id]
  puts "#{doc[:title]} by #{doc[:author]}"
end

index.close
```

Ferret Query Language

Basic Queries

- fox
- w?ldcar*
- quick~0.5
- [20050725 20050905}
- "quick|speedy|fast <> fox"
- "quick|fast fox"~2

Combined queries

- (quick fast) AND fox
- +(qui* fast) -brown +fox
- description:(quick fast) AND name:fox
- content:rails title:rails^5 +date: >= 20070101

Ferret Query API

```
sp_term_q  = SpanTermQuery.new(:content, 'rails')  
sp_first_q = SpanFirstQuery.new(sp_term_q, 100)  
  
result = index.search(sp_first_q)
```

Sorting

```
result = index.search('ruby',  
                      :sort => 'date DESC, title')
```

```
sort = Ferret::Search::Sort.new([  
  Ferret::Search::SortField.new(:date, :reverse => true,  
                                 :type       => :byte)  
  Ferret::Search::SortField.new(:title)  
)  
result = index.search('ruby', :sort => sort)
```

Ferret Browser

[Home](#)[Documents](#)[Terms](#)[Term Vectors](#)[Help](#)

Index Info

Number of Documents 2

Maximum Document 2

Field Infos

Field Name	Boost	stored?	compressed?	indexed?	tokenized?	omit_norms?	store_term_vectors
id	1.0	yes	no	yes	no	no	no
metadata	2.0	no	no	yes	yes	no	yes
name	2.0	no	no	yes	yes	no	yes
product_short_name	2.0	no	no	yes	no	no	yes
product_name	1.0	no	no	yes	yes	no	yes
volume	2.0	no	no	yes	no	no	yes
pdf_content	1.0	no	no	yes	yes	no	yes



+



Basic Usage

```
class Book < ActiveRecord::Base  
  acts_as_ferret  
end
```

```
Book.rebuild_index
```

```
results = Book.find_with_ferret('rails',  
                                :limit => :all)
```

Field Options

```
class Book < ActiveRecord::Base
  acts_as_ferret(
    :fields => {
      :title    => { :store => :yes, :boost => 10 },
      :author   => { :store => :yes },
      :date     => { :index => :untokenized },
      :content  => { }
    },
    :ferret => {
      :analyzer => GermanStemmingAnalyzer.new
    }
  )
end
```

Custom Fields

```
class Author < ActiveRecord::Base
  has_many :books
  acts_as_ferret(
    :fields => {
      :name => { :boost => 10 },
      :book_titles => { }
    }
  )
  def book_titles
    books.map(&:title).join(' ')
  end
end
```

ActiveRecord Options

```
Book.find_with_ferret(  
  query,  
  { :limit => :all },  
  { :conditions => 'in_stock=1',  
    :include    => :author }  
)
```

Lazy Loading

```
class Book < ActiveRecord::Base
  acts_as_ferret(
    :fields => {
      :title    => { :store => :yes },
      :content => { }
    }
  )
end

@results = Book.find_with_ferret query,
                                :lazy => true
```

multi_search

```
Book.multi_search(query, [ Author ],  
                  :limit => :all)
```

```
acts_as_ferret(:fields => { ... },  
              :store_classname => true)
```

Controller

```
def search
  @query = params[:q]
  @results = Book.find_with_ferret(
    @query,
    :per_page => 10,
    :page => params[:page]
  ) unless @query.blank?
end
```

View

```
<% unless @results.blank? %>
  <%= will_paginate @results %>
  <ul>
    <% for r in @results %>
      <li>
        <%= link_to h(r.name), book_path(r) %>
      </li>
    <% end %>
  </ul>
<% end %>
```

Extended Search

```
class BookSearch < ActiveRecord::BaseWithoutTable
  column :query, :string
  column :language, :string

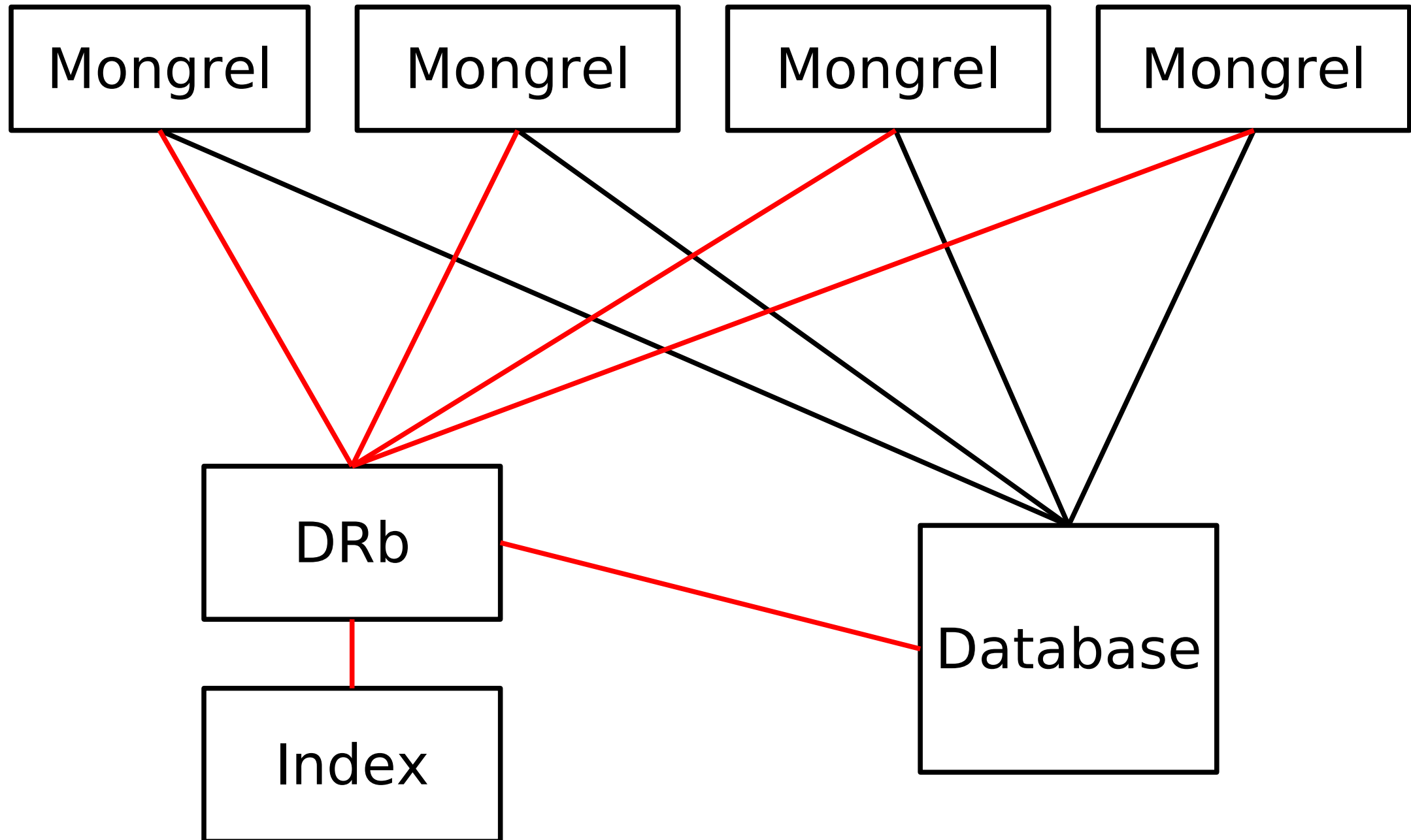
  validates_presence_of :query

  def run(page)
    q = query.dup
    q << " language:#{language}" unless language.blank?
    Book.find_with_ferret q,
                        :per_page => 10,
                        :page => page
  end
end
```

Extended Search

```
def search
  @search = BookSearch.new params[:search]
  if @search.valid?
    @results = @search.run(params[:page])
  end
end
```

Production Mode



Production Mode

config/ferret_server.yml

production:

host: localhost

port: 9010

pid_file: log/ferret.pid

log_level: warn

log_file: log/ferret_server.log

In the Wild

omdb.org

xing.com

blogblogs.com.br

lingr.com

altlaw.org

Peritor Minea (peritor.com)

sachsen-gesetze.de

Further Reading

Ferret Shortcut

Lucene in Action

`ferret-talk@rubyforge.org`

`ferret.davebalmain.com`

`projects.jkraemer.net/acts_as_ferret`

?