

implementing omdb.org

a conclusion after working with rails for more then a year

history of omdb.org

a free movie database that anyone can edit

Brief Description

- omdb (open media database) is a free database for film media
- The source code is available effective today

Timeline

- 11/2005 - Begin of Prototyping (Rails 0.14)
- 03/2006 - Start of 'real' implementation (Rails 1.0)
- 10/2006 - Private Beta Test (Rails 1.0)
- 04/2007 - Public Beta (Rails 1.1.6)

Agenda

- Coding & Conventions
- Internationalization
- Performance
- Maintenance

Coding & Conventions

you're not writing ~~java~~ PHP, you know?

Abstraction (I)

- over abstraction is not a good idea
- use plugins
- write abstract code in plugin style
- write code generators

Abstraction (II)

```
class MovieController < ApplicationController
  include AjaxController
  include TrackBackController

  set_abstract_for :movie
  edit_aliases_for :movie
  image_upload_for :movie
  history_view_for :movie

  lightbox_edit_for :edit_status

  # more code ..
end
```

Abstraction (III)

```
def edit_aliases_for(object, options = {
  :language_independent => false, :allow_independent => false })

  define_method( :edit_aliases ) do
    @item = self.instance_variable_get("@#{object.to_s}")
    @language_independent = true if options[:language_independent]
    @allow_independent = true if options[:allow_independent] and
      not options[:language_independent]
    @aliases = @item.aliases if options[:language_independent]
    render 'common/edit_aliases'
  end

  define_method( :create_alias ) do
    # more code
  end
end
```

Extend Classes (I)

- Ruby classes want to be extended
- don't be shy to change the Rails code

Extend Classes (II)

```
class Movie < ActiveRecord::Base
  acts_as_tree :order => "end, id"
  has_many :people, :through => :casts

  stub [ :casts, :crew, :genres, :plot_keywords,
         :languages ], :if => :empty?
  stub [ :runtime ],
        :if => Proc.new { |value| value == 0 }
end
```

Extend Classes (III)

```
module ActiveRecord
  class Base
    def self.stub( attr_names, opts = {} )
      options = { :if      => :nil?,
                  :clear => false }.merge( opts )
      self.stub_fields = options[:clear] ?
                          {} : ( self.stub_fields ||= {} )
      attr_names.each do |attr|
        self.stub_fields[attr] = options
      end
    end
  end
end
end
```

Extend Classes (IV)

```
class String
  def to_boolean
    (self == 'true' || self == '1') ? true : false
  end

  alias :original_pluralize :pluralize
  def pluralize(count = nil)
    (count && count == 1) ? self : original_pluralize
  end
end
```

Components

don't use them

Internationalization

i18n plugins

- Rails does not support i18n
- choose from one of these plugins
 - globalize
 - gettext / gloc / gibberish

Performance

now that it's done, make it fast

Searching

- DBMS is not an option for complex searches
- use a search engine, like:
 - Ferret / `acts_as_ferret`
 - Solr / `acts_as_solr`
 - Sphinx / `acts_as_sphinx`

Profiling

- read the development.log
- add benchmarks to your methods
- check for memory leaks
 - use plugins like `bleak_house` to find leaks

Caching / Performance

- Ruby is slow
- avoid SQL
- write SQL
- cache everything
 - memcache / `cached_model`
 - page caching / fragment caching

Memcache

```
bash$ script/console
```

```
Loading development environment.
```

```
>> Cache.put :my_key, 'Hello World'
```

```
=> "Hello World"
```

```
>> Cache.get :my_key
```

```
=> "Hello World"
```

```
>> Cache.put :my_other_key, Movie.find(230)
```

```
=> # output ..
```

```
>> Cache.get(:my_other_key).name
```

```
=> "The Godfather"
```

Maintenance

Capistrano

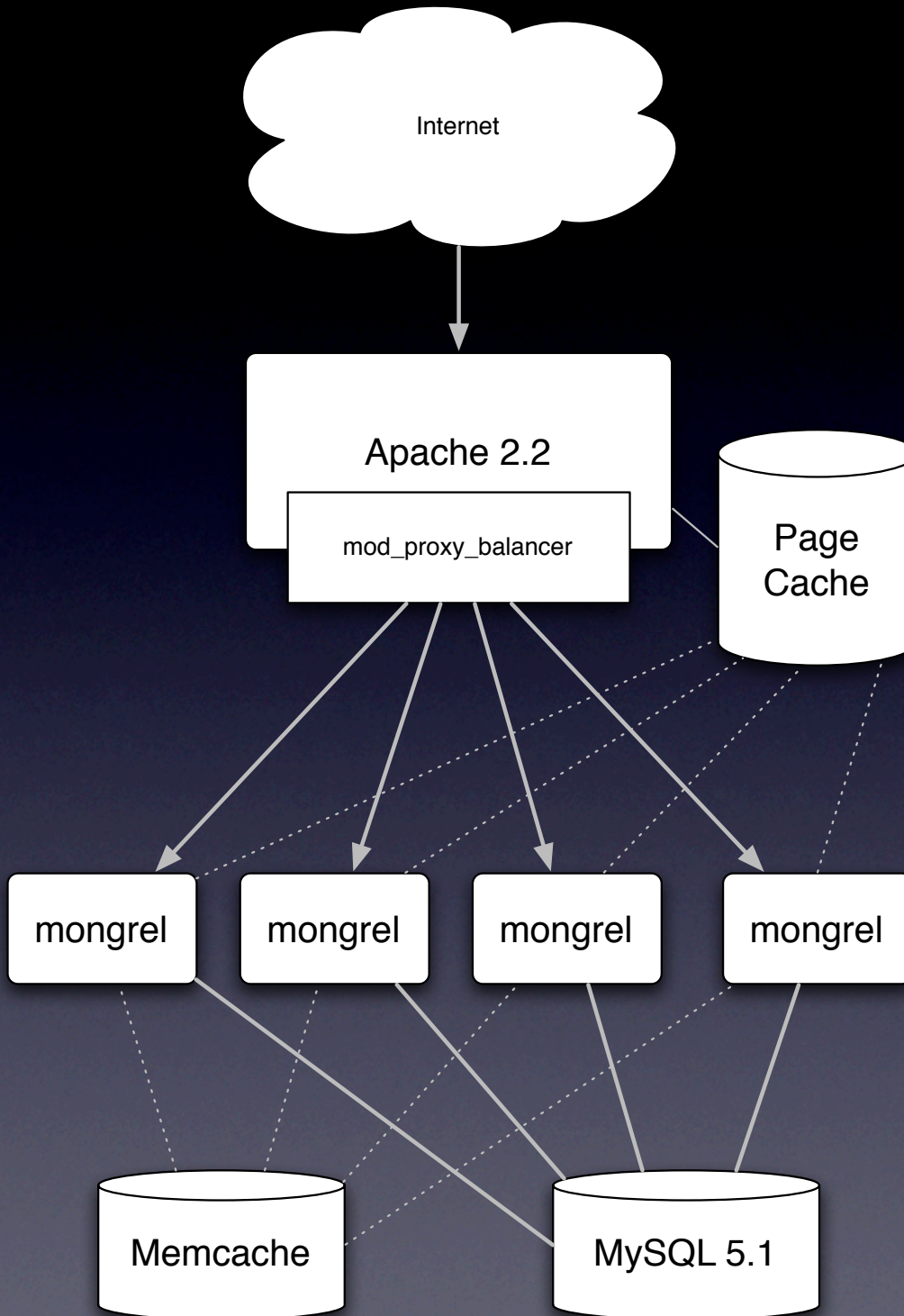
use it

background tasks

- BackgroundDRb - a Ruby job server
 - background indexing
 - cache sweeper
- write rake tasks - schedule via cron

a simple rake task

```
task :select_movie_of_the_day => :environment do
  now = Date.jd(DateTime.now.jd)
  one_year_ago = now - 365
  movie = Movie.find_by_sql("select * from movies where
    (movie_of_the_day < '#{one_year_ago.to_s}' or
    movie_of_the_day is NULL) order by rand()
    limit 1;").first
  movie.movie_of_the_day = now
  movie.save
end
```



Final Thoughts

experiment

participate

contribute

thank you

on behalf of

Benjamin Krause

Mathias Meyer

Thomas Mann

Jens Krämer

Andreas Korth

Ali Jelveh

resources

- <http://bugs.omdb.org/>
- <https://mailings.omdb.org/>
- <http://manuals.rubyonrails.com/read/book/17>
- <http://backgroundrb.rubyforge.org/>
- <http://mongrel.rubyforge.org/>
- <http://ferret.davebalmain.com/trac/>
- <http://www.globalize-rails.org/globalize/>