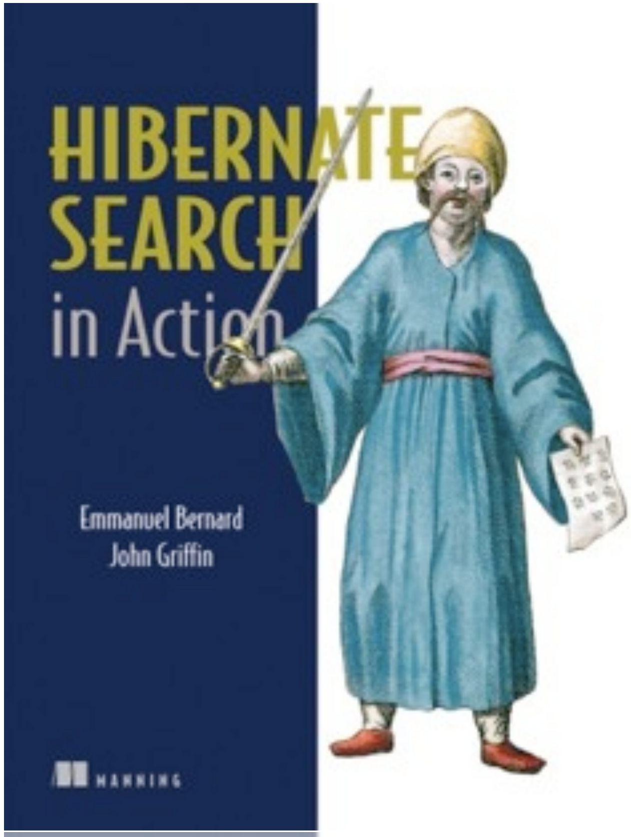
JBoss WORLD
CHICAGO 2009

Hibernate Search:
A Successful Search, a Happy User
Make it Happen!

Emmanuel Bernard
Lead Developer at JBoss by Red Hat
September 2nd 2009

Emmanuel Bernard

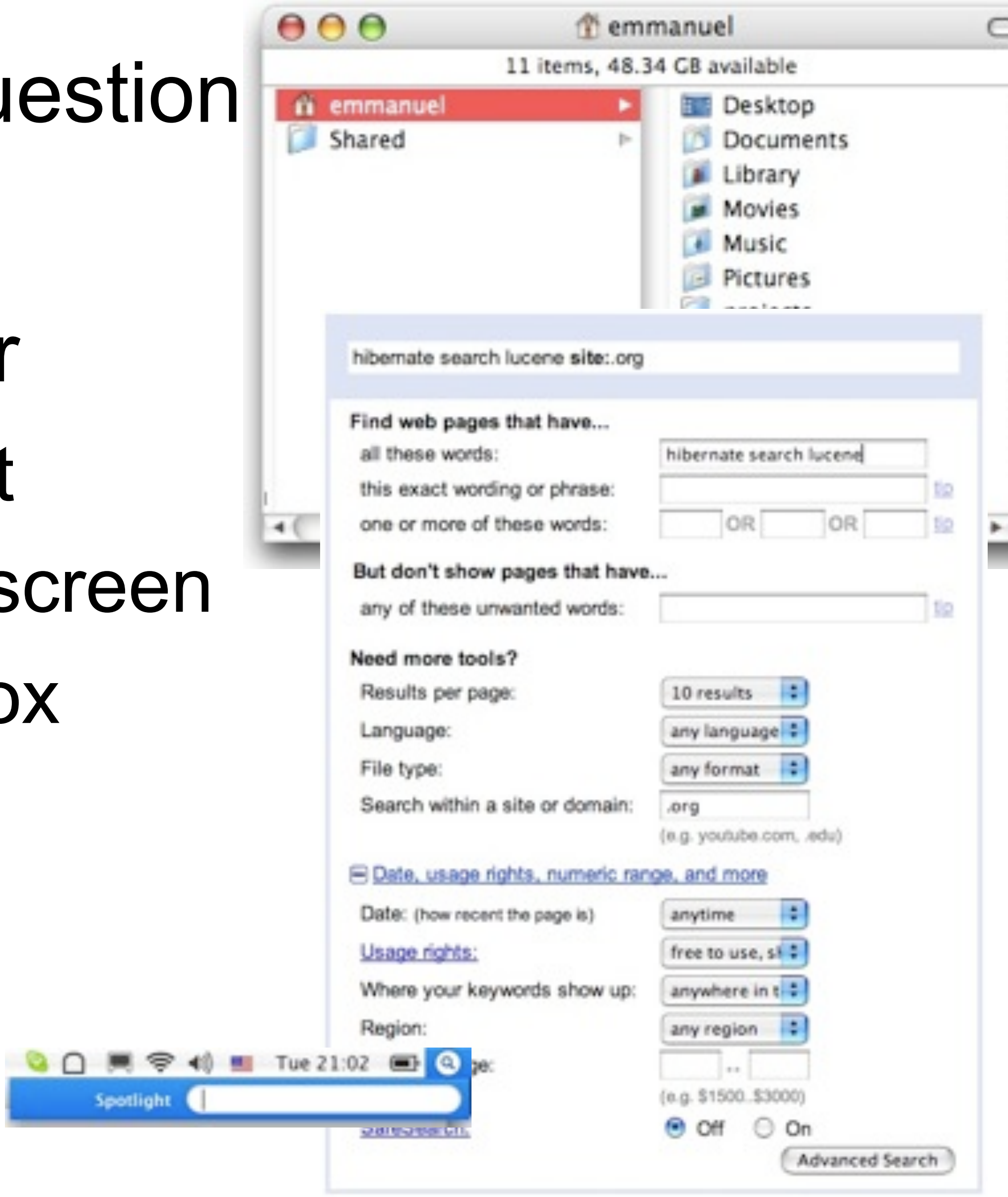
- Hibernate Search in Action
- blog.emmanuelbernard.com
- twitter.com/emmanuelbernard



- Understand what full-text search does for you
- Understand the magic sauce: analyzers
- Full-text search and applications: how does it fit?
- Bring the *Wow!* effect to existing applications

What is searching?

- Searching is asking a question
- Different ways to answer
  - Categorize data up-front
  - Offer a detailed search screen
  - Offer a simple search box



hibernate search lucene **site:**.org

Find web pages that have...

all these words: hibernate search lucene

this exact wording or phrase:

one or more of these words: OR OR

But don't show pages that have...

any of these unwanted words:

Need more tools?

Results per page: 10 results

Language: any language

File type: any format

Search within a site or domain: .org

(e.g. youtube.com, .edu)

Date, usage rights, numeric range, and more

Date: (how recent the page is) anytime

Usage rights: free to use, sh

Where your keywords show up: anywhere in t

Region: any region

(e.g. $1500..$3000)

● Off ○ On

Human search
in a relational world

- where?
  - which columns, which tables
- column != word
  - wildcard queries?
- did you say “car” or “vehicle”?
- cympausium or simposyum?
- Order results by relevance

- How to do that in SQL?

Full Text Search

- Search by word
- Dedicated index
  - inverted indices (word frequency, position)
- Very efficient

- Full text products:
  - embedded in the database engine
  - black box / appliance
  - library embeddable like Lucene

Some of the interesting problems

- bring the “best” document first
- recover from typos
- recover from faulty orthography
- find from words with the same meaning
- find words from the same family
- find an exact phrase
- find similar documents

Find by relevance

- Best results first
  - very human sensitive
- Prioritize some fields over others
- The more matches, the better
  - for a given key word per document
  - for a given document the amount of matching key words
- Similarity algorithm

Extracting the quintessence

- Word: Atomic information

- Analyzer
  - Chunk / tokenize the text into individual words
  - Apply filters
    - remove common words
    - lower case
- One tokenizer
- Some filters

Approximation

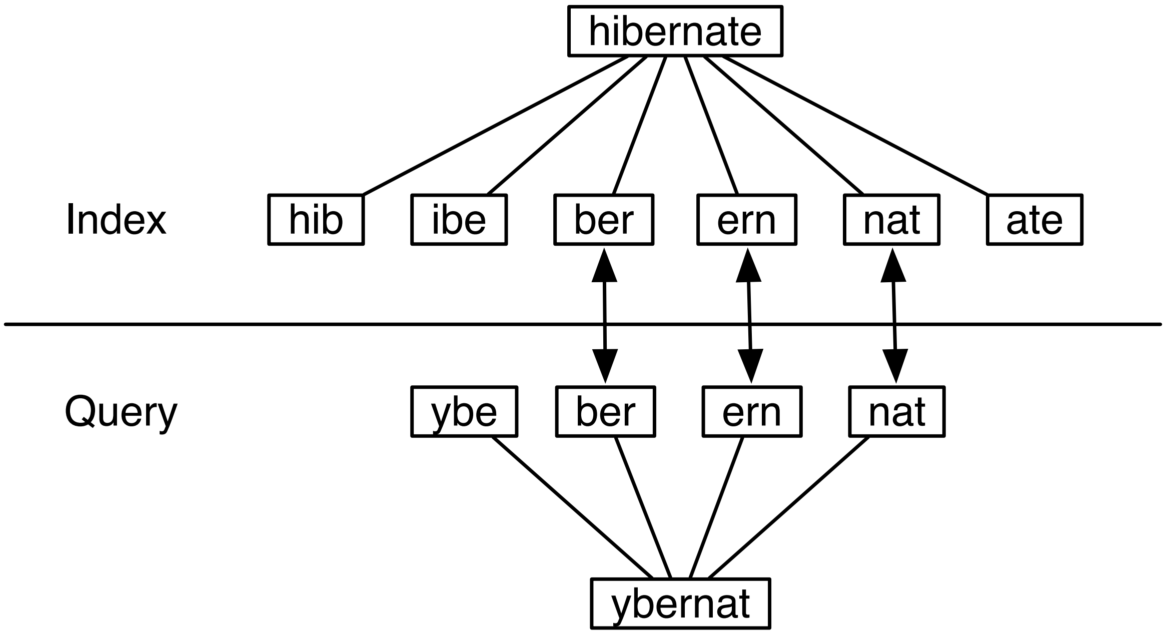
- Recover from typos and other approximations

- Fuzzy search
  - query time operation
  - Levenshtein distance (edit distance)

```
Hibernate

Hibrenate
```

- n-gram
  - cut the word in parts of n characters
  - index each piece



- Indexing + query
  - use a TokenFilter

Demo

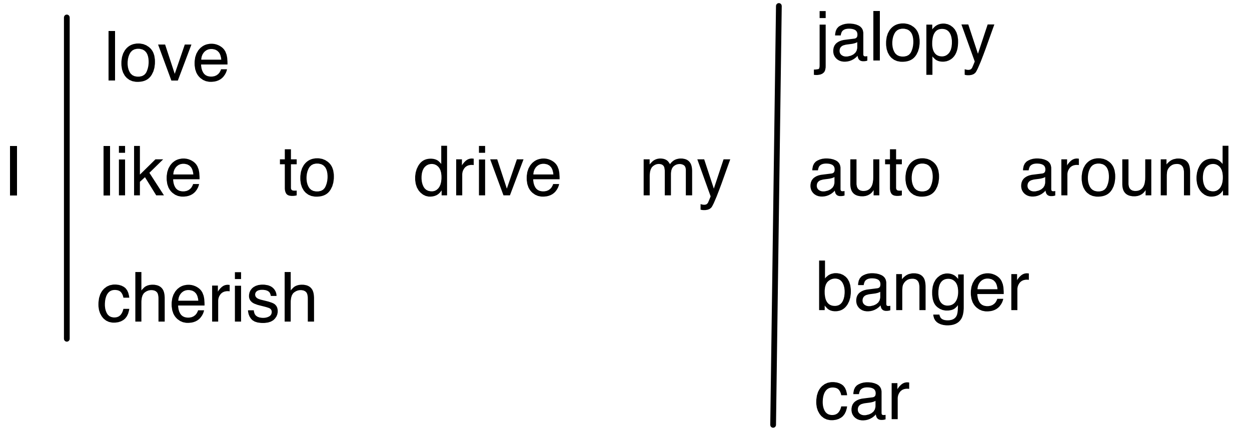
Phonetic search

- Is it “jiroscop” or “gyroscope”
  - not so useful in daily life
- Several phonetic algorithms
  - Soundex
  - Metaphone (JRSKP)
  - mostly for latin languages
- index the phonetic equivalent of a word

- Indexing + query time strategy
  - use a TokenFilter

Synonyms

- Based on a synonym dictionary
- index all synonyms of a word in the index



love

I like to drive my auto around

cherish

jalopy

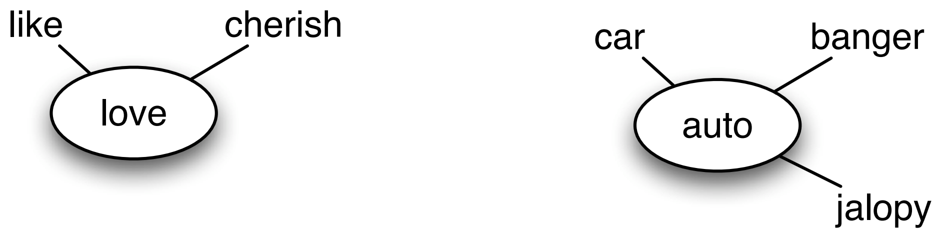
banger

car

- Indexing time strategy
  - use a TokenFilter

Synonyms

- Based on a synonym dictionary
- index a reference word in the index



like

cherish

love

car

banger

auto

jalopy

I like to drive my auto around

I love to drive my banger around

I cherish to drive my car around

I love to drive my auto around

- Indexing + query time strategy
  - use a TokenFilter

Words from the same family

- love, lover, loved, loving
- Brutal force
  - index all variations of a word
- Stemming
  - Porter algorithm for English
  - Snowball Stemmer for most Indo-European languages
- Indexing + query time strategy
  - use a TokenFilter

Demo

What’s the catch

- Lucene is quite low level
- Integration into an application model
- Index synchronization
- Object model conversion
- Programmatic mismatch

Integration into a Java SE / EE app

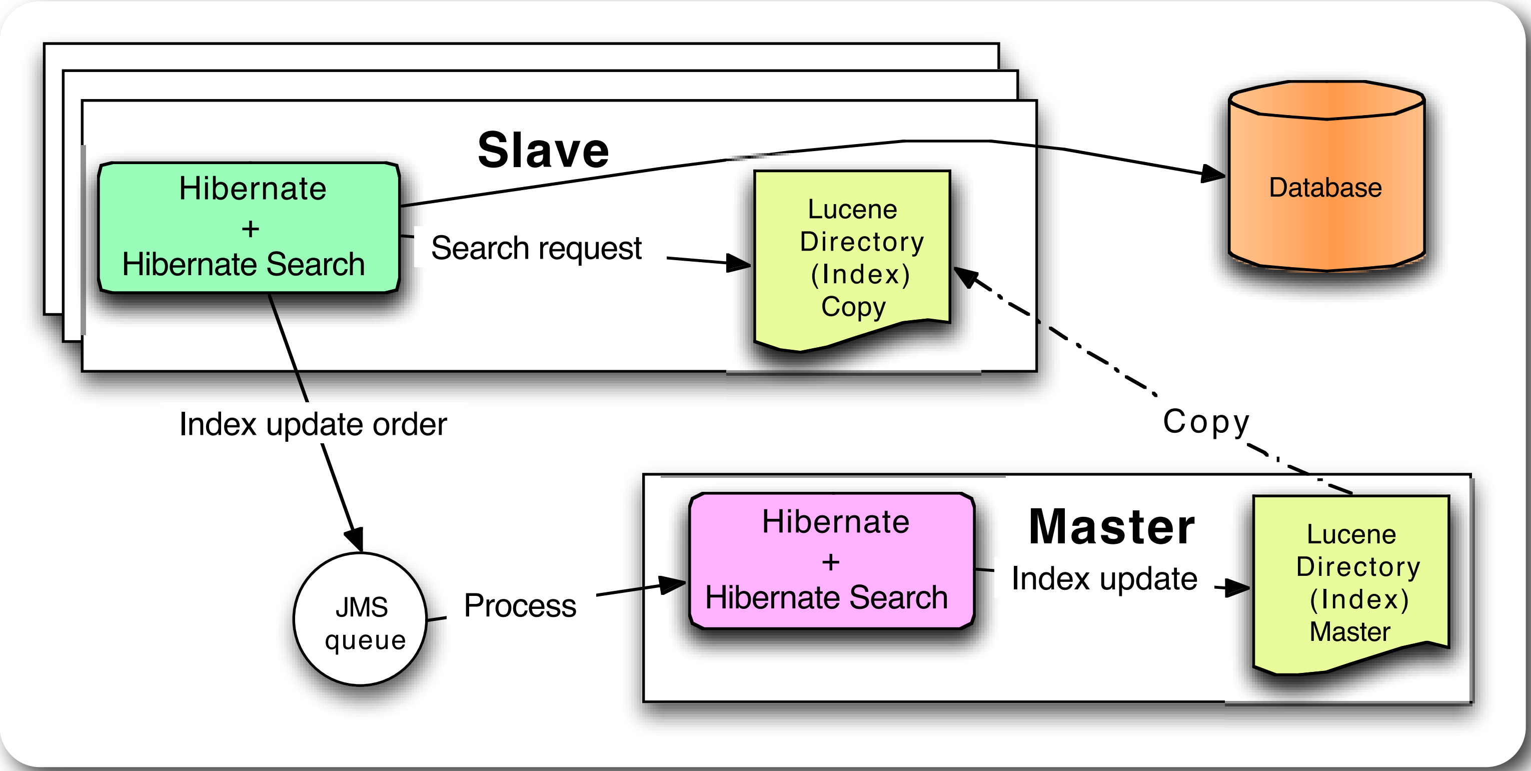
- Hibernate Search bridges
  - Hibernate Core and Java Persistence
  - Apache Lucene
- Transparent index synchronization
  - event based
- Metadata driven conversion
  - annotation based
- Unified programmatic model
  - API
  - semantic

More on Hibernate Search

- Asynchronous clustering
- Projection
- Filters
- Index sharding
- Custom DirectoryProvider (eg. JBoss Cache based)
- JBoss Cache is full text searchable
- Native Lucene access

Asynchronous cluster

- Search local / change sent to master
- Asynchronous indexing (delay)
- No front end extra cost / good scalability

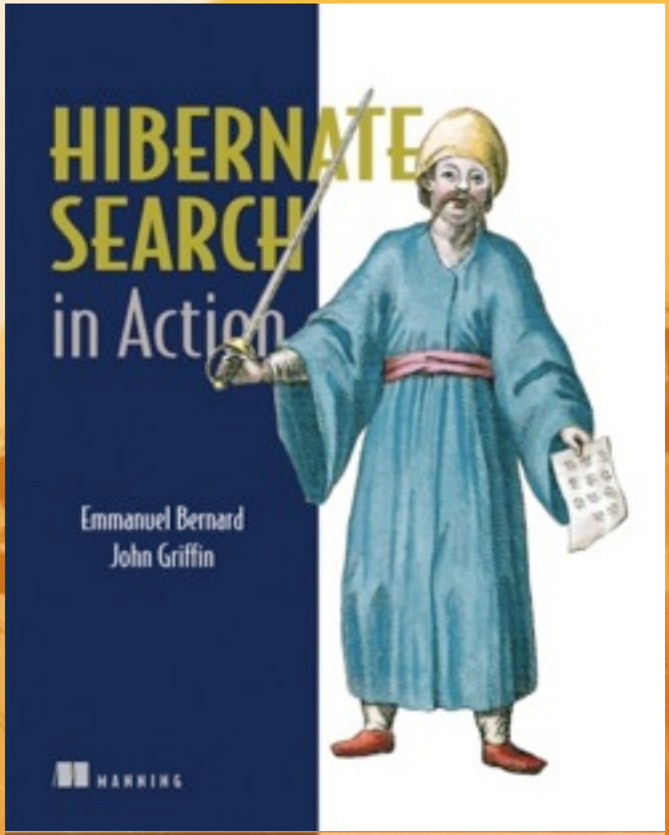


Slave
Hibernate + Hibernate Search
Search request
Lucene Directory (Index) Copy
Database
Index update order
Copy
JMS queue
Process
Master
Hibernate + Hibernate Search
Index update
Lucene Directory (Index) Master

Summary

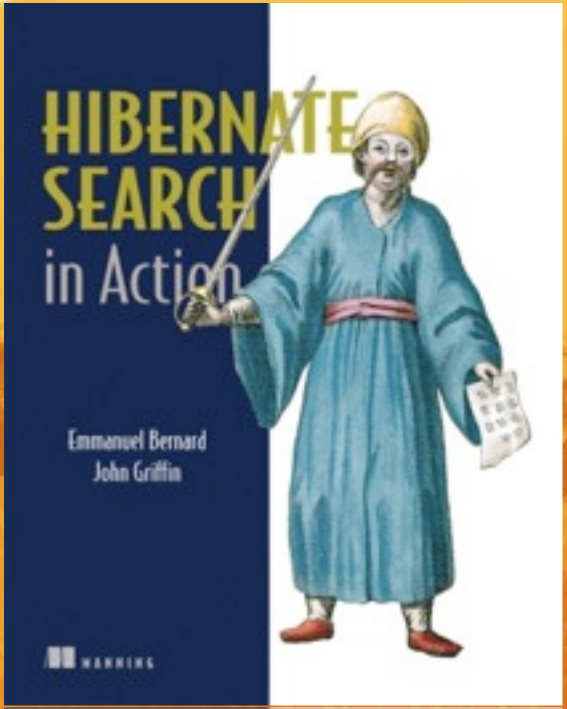
- Search for humans
- Full text tackles those problems
  - relevance
  - (human) fault tolerance
  - stemming and synonyms
  - incremental search
- Barrier of entry has lowered: Go for it!
  - POJO based approach
  - infrastructural code tackled by frameworks
  - unified programmatic model

Questions


HIBERNATE SEARCH in Action
Emmanuel Bernard
John Griffin
MANNING

For More Information

- http://search.hibernate.org
- http://lucene.apache.org
- Hibernate Search in Action
  - Manning



HIBERNATE SEARCH in Action

Emmanuel Bernard
John Griffin

MANNING

- http://in.relation.to
- http://blog.emmanuelbernard.com

QUESTIONS?

TELL US WHAT YOU THINK:
REDHAT.COM/JBOSSWORLD-SURVEY