Hibernate Search

Full-text search for Hibernate applications

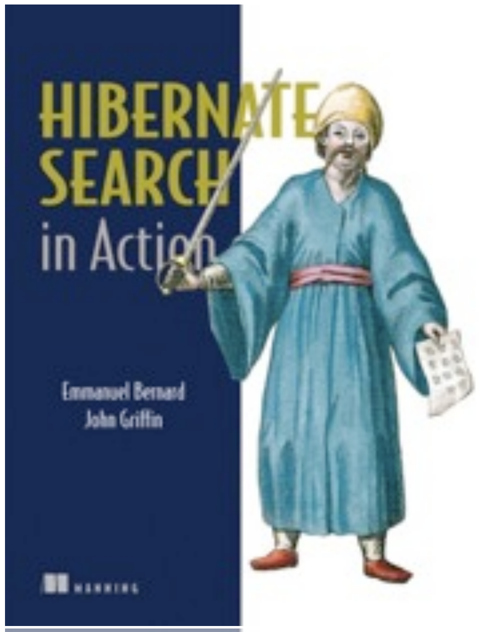
Emmanuel Bernard
JBoss, by Red Hat
http://in.relation.to/Bloggers/Emmanuel

- 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

Emmanuel Bernard

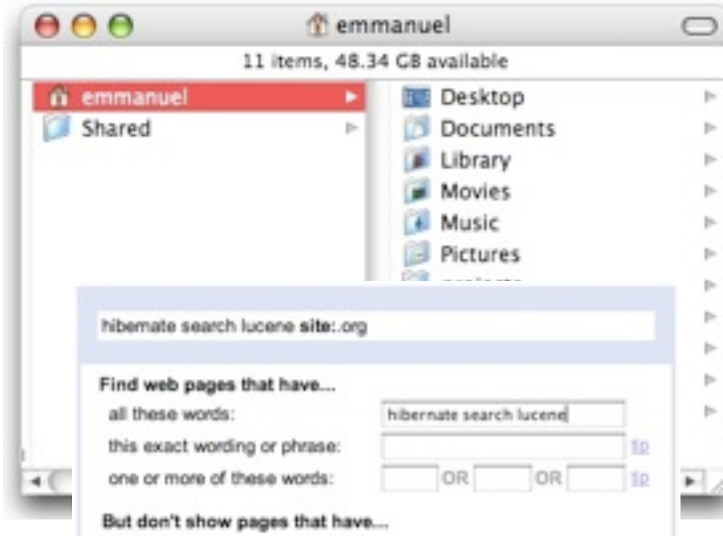
- Hibernate Search in Action
- blog.emmanuelbernard.com
- twitter.com/emmanuelbernard
- lescastcodeurs.com

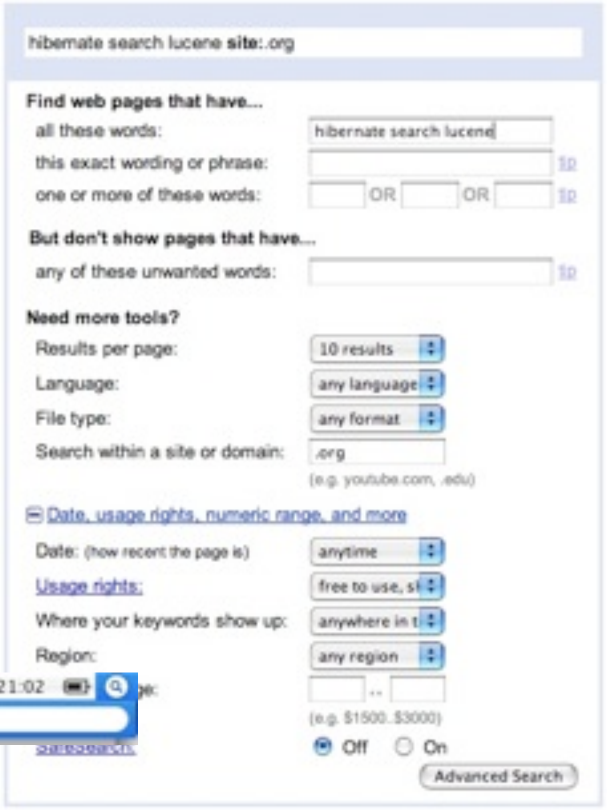
Les CastCodeurs
"En Français dans le Code"
http://lescastcodeurs.com


HIBERNATE SEARCH in Action
Emmanuel Bernard
John Griffin
MANNING

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, s
Where your keywords show up:	anywhere in t
Region:	any region

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

[✓ Off] On

Advanced Search

Human search in a relational DB

- where? (which columns, which tables)
- column != word (wildcard queries?)
- did you say “car” or “vehicle”?
- konference or conpherance?
- 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
  - 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

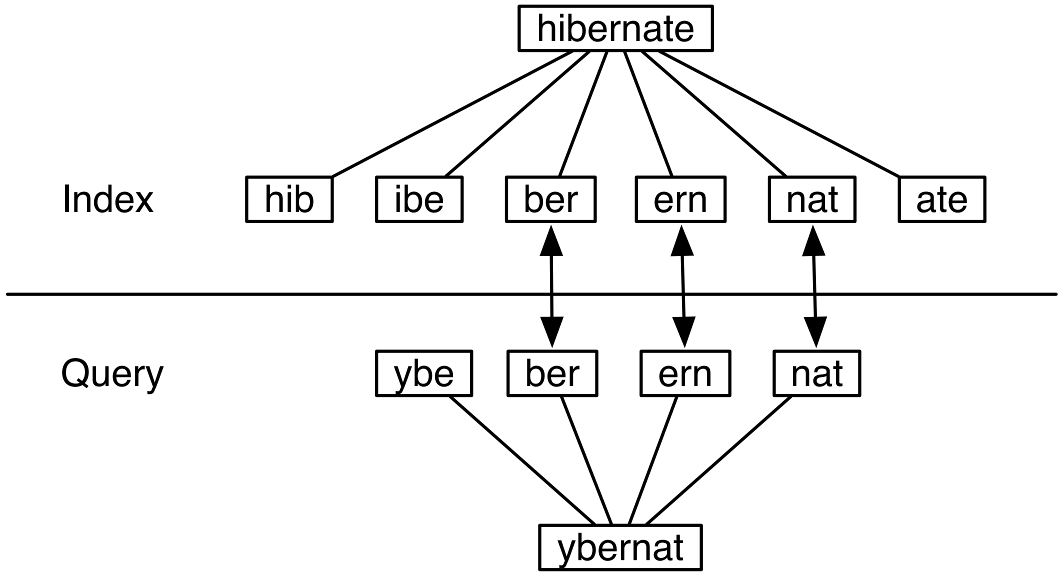
Approximation

- 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 time strategy

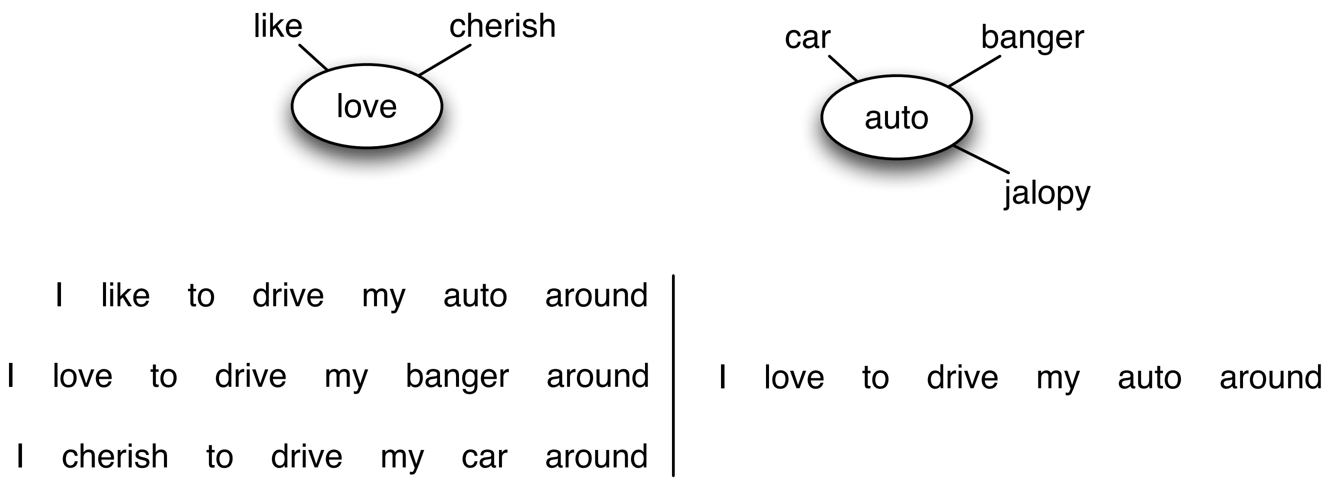
Demo

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 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
- 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 in Java SE / EE

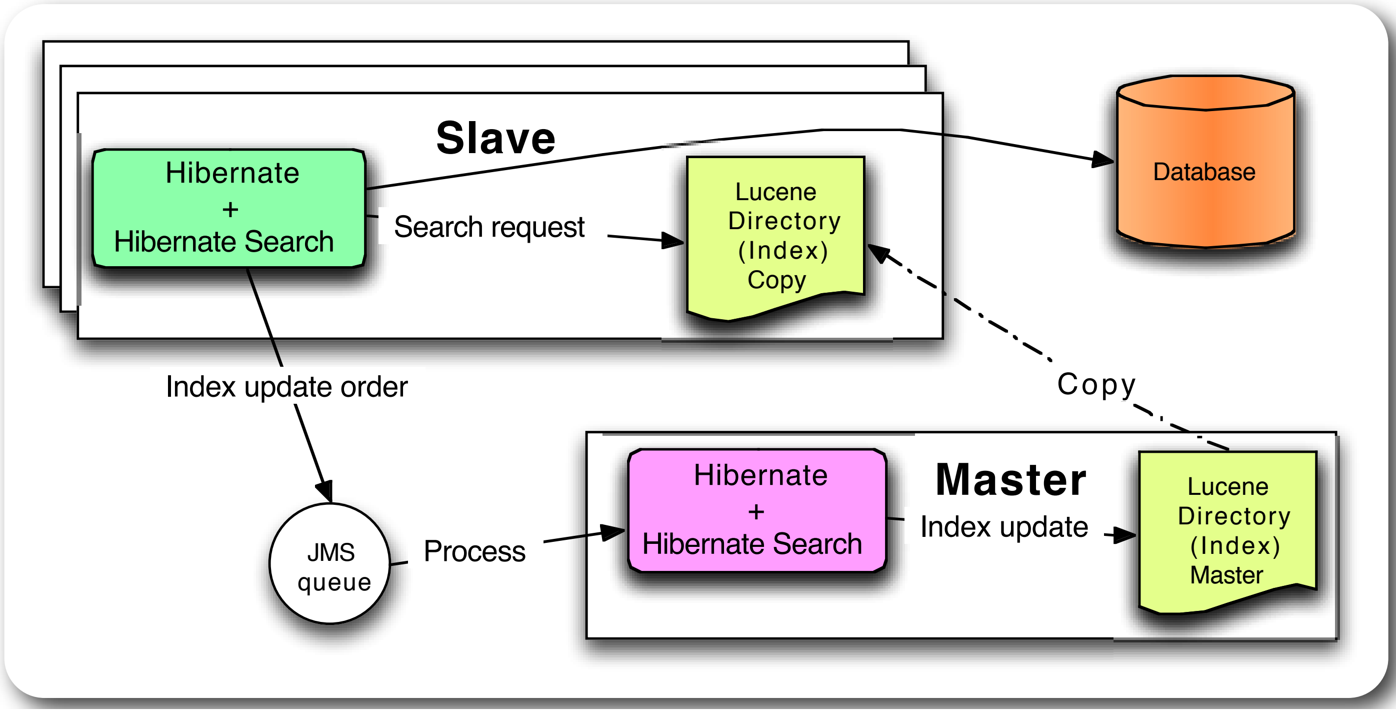
- Hibernate Search bridges
  - Hibernate Core and Java Persistence
  - JBoss Cache & Infinispan (More to come)
  - Apache Lucene
- Transparent index synchronization (event based)
- Metadata driven conversion (annotation based)
- Unified programmatic model
  - API
  - semantic

More on Hibernate Search

- Asynchronous clustering (JMS, JGroups)
- Projection
- Filters
- Index sharding
- Custom DirectoryProvider (eg. JBoss Cache, Infinispan based)
- Infinispan / JBoss Cache 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
Hibernate + Hibernate Search
Master
Index update
Lucene Directory (Index) Master

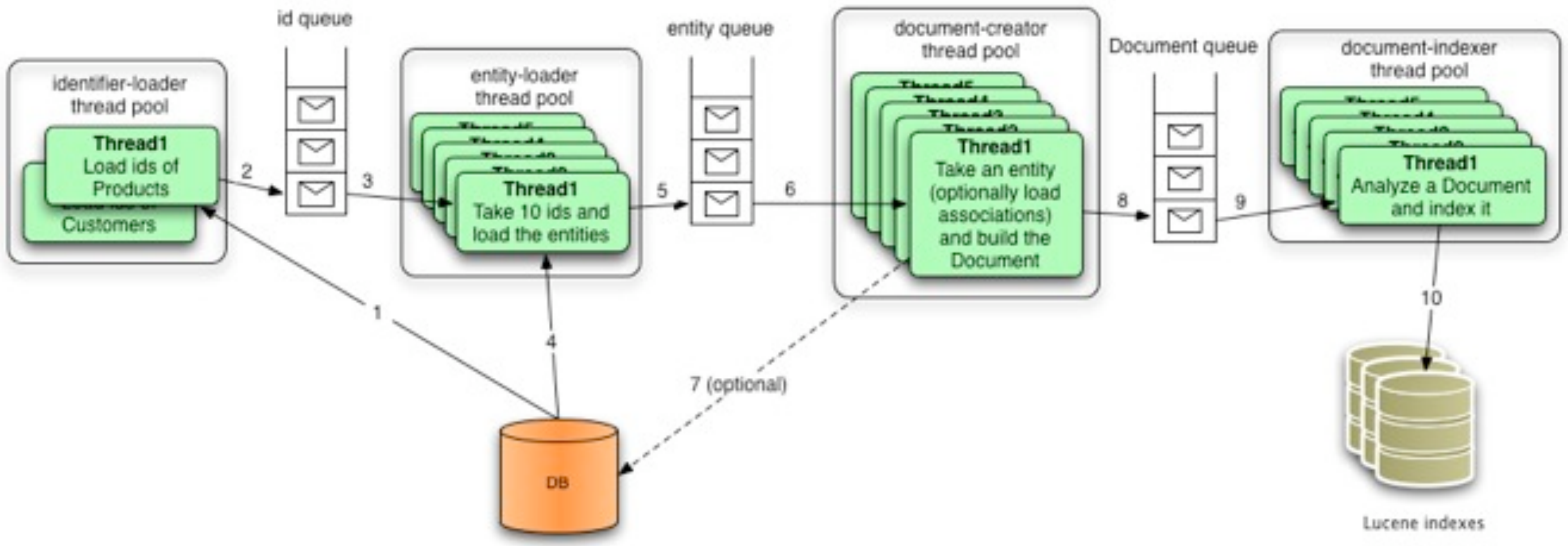
Summary

- 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

Future

- Perf
  - on mass indexing
  - on simple deployments
- Ease of use
  - programmatic API to express mappings
  - fluent API to write queries
- Cluster
  - JGroups based back end
  - Infinispan based Lucene directory

Massive indexing



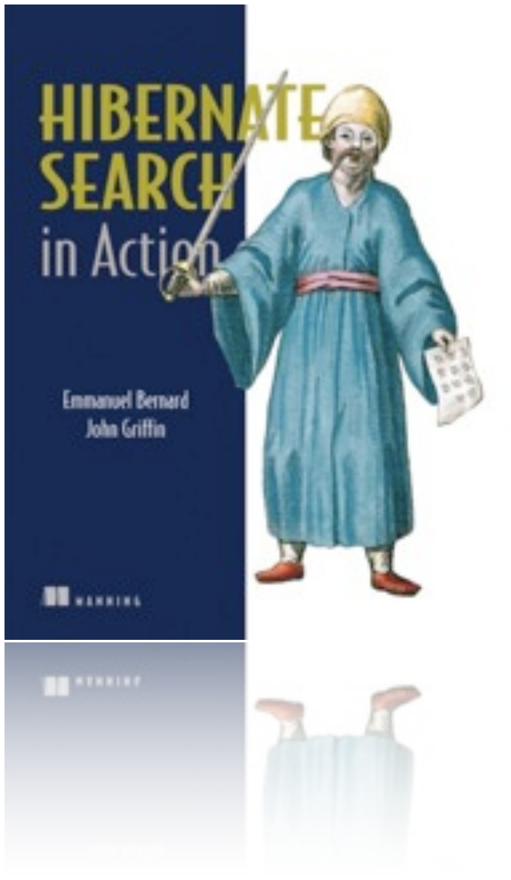
id queue
identifier-loader thread pool
Thread1
Load ids of Products
Customers
2
3
entity-loader thread pool
Thread1
Take 10 ids and load the entities
5
entity queue
6
document-creator thread pool
Thread1
Take an entity (optionally load associations) and build the Document
8
Document queue
9
document-indexer thread pool
Thread1
Analyze a Document and index it
10
1
4
7 (optional)
DB
Lucene indexes

Programmatic configuration

```
SearchMapping mapping = new SearchMapping();
      mapping.analyzerDef( "stem", StandardTokenizerFactory.class )
               .tokenizerParam( "name", "value" )
               .tokenizerParam(  "name2", "value2" )
               .filter( LowerCaseFilterFactory.class )
               .filter( SnowballPorterFilterFactory.class)
                  .param("language", "English")
            .entity(Address.class).indexed().indexName("Address_Index")
               .property("street1", ElementType.FIELD)
                  .field()
                  .field()
                     .name("street1_iso")
                     .store( Store.YES )
                     .index( Index.TOKENIZED )
                     .analyzer( ISOLatin1Analyzer.class)
                  .field()
                     .name("street1_ngram")
                     .analyzer("ngram")
            .entity(User.class).indexed()
               .property("name", ElementType.METHOD)
                  .field()
            .analyzerDef( "minimal", StandardTokenizerFactory.class  );
```

Questions

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



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