

# 3 Use Cases for eZ Find



by Ivo Lukač  
[www.netgen.hr/eng](http://www.netgen.hr/eng)

## 3 Use cases for eZ Find

1. Suggestion extension based on eZ Find
2. Increasing template performance with eZ Find
3. Geo search with eZ Find

## Why use eZ Find / Solr?



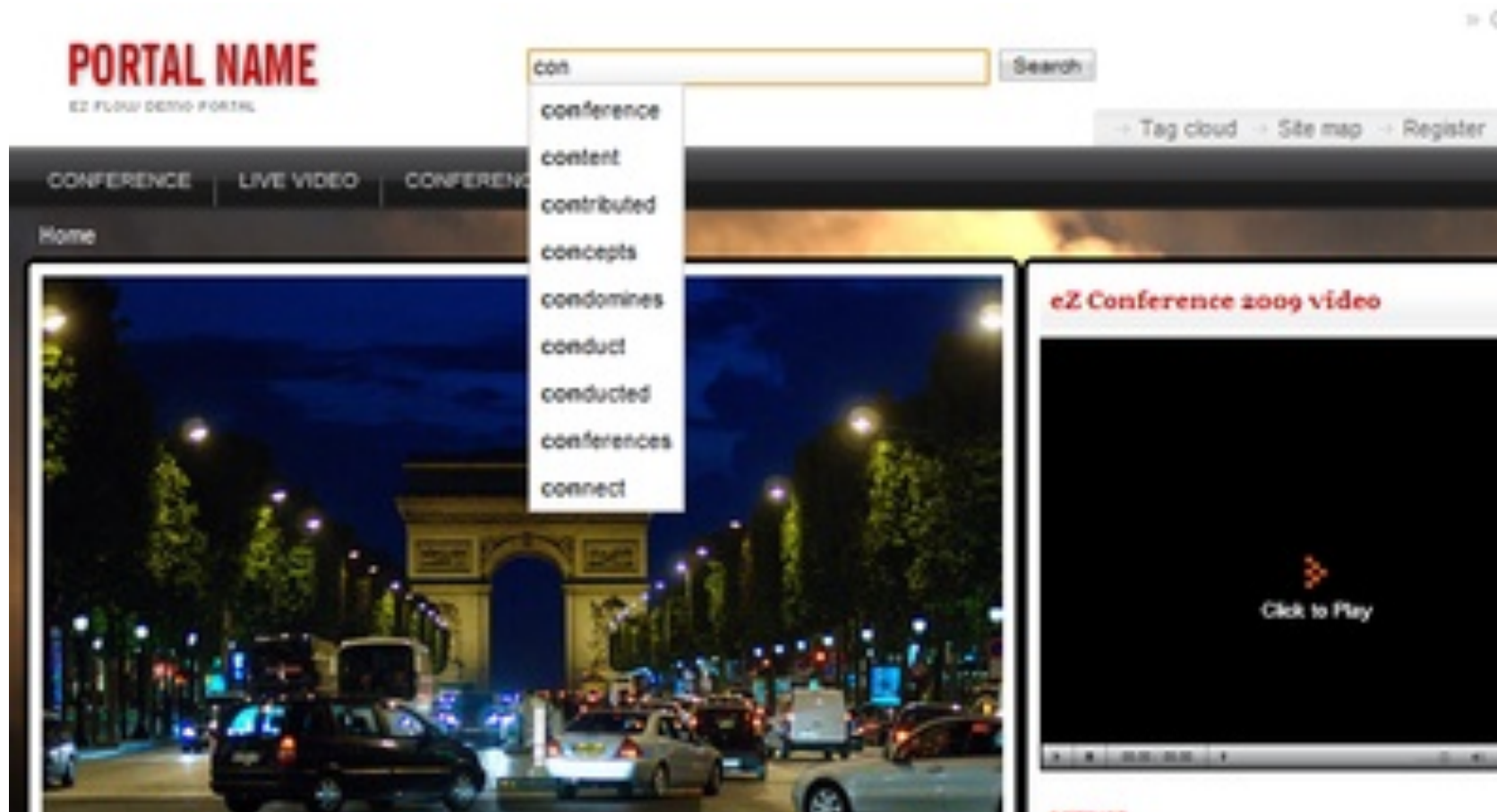
# Case 1 – Suggestions while searching

## TASK

- Drop down list of suggestions when typing text in search field
- Needs to be based on content data
- Around 300 000 objects
- Needs to be fast  
(users are familiar with Google)



# Case 1 – Suggestions while searching



## Case 1 – Suggestions while searching

### SOLUTION

- MySQL queries with LIKE would be too slow
- Develop a simple extension using solr facets query with prefix parameter (eZ Find index was already in place)

## Case 1 – Suggestions while searching

### SOLR QUERY PARAM 1

- rows=0 (standard search results are not needed)
- facet=on (enable facets)
- facet.field=ngsuggest\_text (facet field defined in ngsuggest.ini)
- facet.prefix=con (letters user entered)
- facet.limit=10 (maximal number of suggestions)

## Case 1 – Suggestions while searching

### SOLR QUERY PARAM 2

- facet.mincount=1 (minimum occurrences for word to show up in suggestions)
- q=\*:~\* (all index is searched for the first word; previous words searched for second, third, etc)
- fq=meta\_is\_hidden\_b:false+AND  
+meta\_is\_invisible\_b:false (filter conditions: root node id, classes, installation id, .. )
- wt=json (return results in JSON format)



## Case 1 – Suggestions while searching

### EZ PUBLISH MODULE

- Proxy module (no direct call to Solr)
- Builds query from user input and ini settings
- Just outputs JSON results from solr

## Case 1 – Suggestions while searching

### CLIENT SIDE

- jQuery JSON suggestsearch component
- Calls eZ suggestion module over AJAX  
/ngsuggest/searchsolr?id=[search\_id]&keyword=[text]
- Uses returned JSON to show the list

## Case 1 – Suggestions while searching

TRY IT OUT NOW

<http://projects.ez.no/ngsuggest>

Follow INSTALL.txt

## Case 2 – Increasing template performance

### SITUATION

- Big site with lots of logged in users
- Lot of fast changing content
- Standard caching method not enough
- Templates need to be faster



## Case 2 – Increasing template performance

### SOLUTION

- Replacing content list/tree function with ezfind search
- Fetch will be faster
- Database will have less queries

## Case 2 – Increasing template performance

### BASIC NOTIONS

- DelayedIndexing ?
- OptimizeOnCommit=disable
- AllowEmptySearch=enable
- Not all data is in the index

## Case 2 – Increasing template performance

### FEATURES OF EZFIND FETCH

- Text searching
- Offset, limit, sort
- Filter by class, section, subtree, etc.
- Ignoring visibility
- Limitation

## Case 2 – Increasing template performance

### REPLACING TREE FETCH

- Use subtree\_array parameter

### REPLACING LIST FETCH

- Use filter -> e.g. main\_parent\_node\_id:2



## Case 2 – Increasing template performance

### REPLACEMENT1

Content list/tree	eZ Find search
<code>parent_node_id</code>	<code>subtree_array</code>
<code>sort_by</code>	<code>sort_by</code> (no priority, no depth)
<code>offset &amp; limit</code>	<code>offset &amp; limit</code>
<code>attribute_filter</code>	<code>filter</code> (different format, more options)

## Case 2 – Increasing template performance

### REPLACEMENT 2

Content list/tree	eZ Find search
<code>extended_attribute_filter</code>	filter with CopyFields (less indexed data → less capable)
<code>class_filter_type</code> & <code>class_filter_array</code>	<code>class_id</code> (excluding only through <code>ezfind.ini</code> )
<code>only_translated</code> & <code>language</code>	filter ( <code>language_code:ger-DE</code> , <code>SearchMainLanguageOnly</code> )

## Case 2 – Increasing template performance

### REPLACEMENT 3

Content list/tree params	eZ Find search params
<code>main_node_only &amp; depth</code>	No data in index
<code>as_object</code>	Not implemented yet
<code>limitation</code>	<code>limitation</code>
<code>ignore_visibility</code>	<code>ignore_visibility</code>

## Case 2 – Increasing template performance

### AND BEYOND

- Text search
- Facets
- Boosting
- Highlighting
- Spell checking

## Case 2 – Increasing template performance

### BENEFITS

- Speed, speed & speed
- Score/relevance ranking
- Additional functionalities
- Shards for multilanguage sites
- No need for extra count fetch
- More powerful filtering



## Case 2 – Increasing template performance

### TRADEOFFS

- Data dependent on the indexing
- Some sort methods missing
- No main\_node\_only switch
- No class excluding (only in ini)
- Filtering depends on what data is indexed

## Case 2 – Increasing template performance

### MORE DETAILS

<http://share.ez.no/tutorials>

### FUTURE

- In next versions of eZ Find all object data will be indexed
- Even less SQL queries needed
- Attribute storage

## Case 3 – Geo Search

### TASK

- Make a mobile interface to search for nearest locations
- Use phone location as a reference
- Needs to be fast 😊





## Case 3 – Geo Search

### PREREQUISITES

- eZ Find 2.2 with geopoint field type
  - LatLon field type in future versions
- ezgmaplocation extension
- ezfSolrDocumentFieldGmapLocation class for indexing
- Problem with dash “-”

## Case 3 – Geo Search

### EUCLIDIAN DISTANCE

- $\text{dist}() = \text{sqrt}((\text{lat1} - \text{lat2})^2 + (\text{lng1} - \text{lng2})^2)$

### WE DON'T NEED THE SQRT()

- $\text{sqedist}() = (\text{lat1} - \text{lat2})^2 + (\text{lng1} - \text{lng2})^2$

### BOOSTING WITH DISTANCE

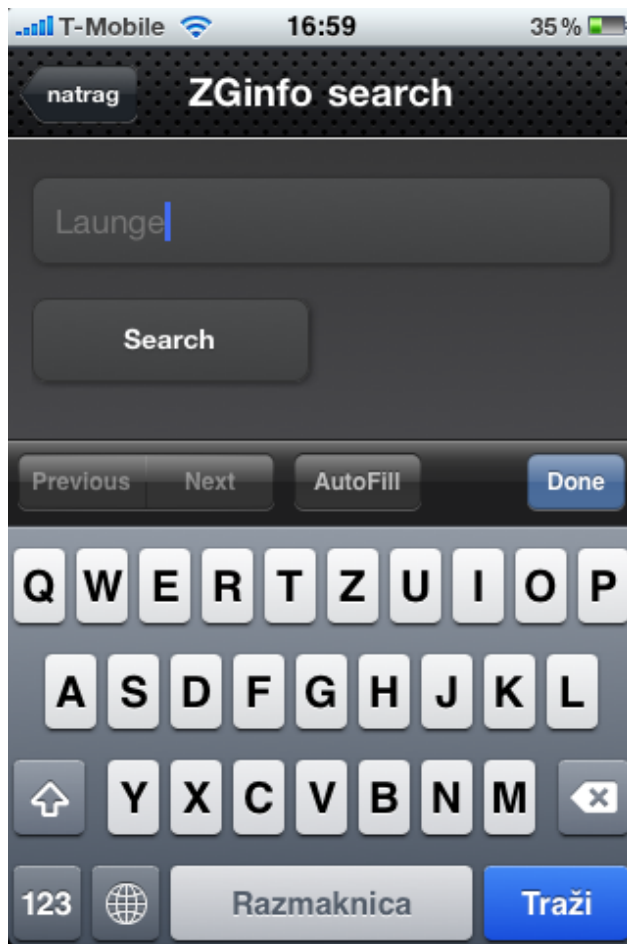
- $\text{recip}(\text{sqedist}(2, \text{point1}, \text{point2}), 1, 1, 0)$

## Case 3 – Geo Search

### QUERY TIME BOOSTING

- Either concat next string with search term:
  - `_val_:"recip(sqedist(gmaps_coordinates,vector(48.166085,-104.326172)),1,1,0)"`
- Or use 'boost\_functions' param with:
  - `hash('functions',array('recip(sqedist(gmaps_coordinates,vector(48.166085,-104.326172)),1,1,0)'))`

## Case 3 – Geo Search



## Case 3 – Geo Search

MORE DETAILS

<http://www.netgen.hr/eng/Blog>

## 3 Use Cases for eZ Find

QUESTIONS ?