Lincolm

Lucene HitCollector implementation with FieldCache to create document categorized searches with minimal impact to performance.

When creating a lucene search application, one commom problem is create a categorized search with groupings count based on that categories. Suppose you want a search grouping documents found by a specific criteria. You have a list of articles, this articles has several fields (title, body, theme, date, keywords, and on). Now, you want to create a search over this articles list (materialized do a lucene index) and the search result must show a grouping count for every article found on every theme for all document set.

Using some specialization to default HitCollector implementation can be easily done, and will allow you to iterate over all document set. Iterate over all document set has a cost, but, in this case it's necessary due search requisit: create grouping counts for every theme found in a given search.

Access the disk, same using a FieldSelector it's not the best way because increase search response time due much more IO activity to read documents on index. One efficient way I found was iterate over all document set, but, instead of read all document from index (even using a FieldSelector), I read the theme information from FieldCache implementation. So, the query runs faster because no IO access is necessary!

Another problem I found was if an article has a multivalued field. The default FieldCache implementation returns the first field value, only. So, I created a specialized FieldCache in which a store the multivalued field as TermFreqVector.YES and I used IndexReader.getTermFreqVector(doc) to read back the field array. During searcher creation - new IndexSearcher(reader) - I populate this special FieldCache and I used it directly from my HitCollector.

My application is running on production environment since 01/Jan/2009 at address: http://www.clicrbs.com.br/busca/rs (portuguese site). So, a sample searching, grouping and creating paging can be found like this: http://www.clicrbs.com.br/busca/rs?c=-1&q=tipo%3Amaterias&t=2009 (end user mode) http://www.clicrbs.com.br/busca/rs?c=-1&debug=true&q=tipo%3Amaterias&t=2009 (debug mode with elapsed time)