<http://mail-archives.apache.org/mod_mbox/lucene-dev/200110.mbox/%3C3BC755AA.9070106@earthlink.net%3E>

From dmitrys@earthlink.net Fri Oct 12 20:42:32 2001

Return-Path: <dmitrys@earthlink.net>

Mailing-List: contact lucene-dev-help@jakarta.apache.org; run by ezmlm

Delivered-To: mailing list lucene-dev@jakarta.apache.org

Received: (qmail 38378 invoked from network); 12 Oct 2001 20:42:32 -0000

Received: from relay3.uswest.net (HELO relay1.uswest.net) (63.226.138.11)

by daedalus.apache.org with SMTP; 12 Oct 2001 20:42:32 -0000

Received: (qmail 3491 invoked by uid 0); 12 Oct 2001 20:42:33 -0000

Received: from unknown (HELO earthlink.net) (65.100.117.194)

by relay3.uswest.net with SMTP; 12 Oct 2001 20:42:33 -0000

Message-ID: <3BC755AA.9070106@earthlink.net>

Date: Fri, 12 Oct 2001 14:42:18 -0600

From: Dmitry Serebrennikov <dmitrys@earthlink.net>

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:0.9.2) Gecko/20010726 Netscape6/6.1

X-Accept-Language: en-us

MIME-Version: 1.0

To: lucene-dev@jakarta.apache.org

Subject: Some success reports

Content-Type: text/plain; charset=us-ascii; format=flowed

Content-Transfer-Encoding: 7bit

X-Spam-Rating: daedalus.apache.org 1.6.2 0/1000/N

Greetings,

First, some success reports:

- Made a few Searcher methods public that were previously package-visible:

int maxDoc() and docFreq(Term)

This allows easier creation of BitVectors to contain all documents

without calling IndexReaders directly. While you typically have access

to IndexReader for an IndexSearcher, MultiSearcher forces you to

maintain a whole list of them if you need direct access. This way,

everything can be done from the Searcher / MultiSearcher interface itself.

- Added a few methods to Searcher for efficient counting of search results:

Searcher.count(Query query)

Searcher.count(Query query, Filter filter, float minScore)

For the minScore, only documents that have the score higher than

minScore are counted. This is implemented in Searcher by constructing a

BitVector and marking off each found document that meets the criteria,

then returning the count. For a single IndexSearcher it is at least

twice as fast as running a Query and then getting the total count from

the Hits object. For a MultiSearcher, the speed gain is not as

impressive yet. I think I need to have a more optimized implementation

for it. Working on that.

Any objections to the above changes? I'll send in the diffs when I'm

done and the objections / suggestions are incorporated.