http://mail-archives.apache.org/mod\_mbox/lucene-dev/200110.mbox/%3c4BC270C6AB8AD411AD0B00B0D0493DF0EE7CBE@mail.grandcentral.com%3e

From lucene-dev-return-290-qmlist-jakarta-lucene-dev-archive=jakarta.apache.org@jakarta.apache.org Mon Oct 29 18:09:32 2001

Return-Path: <lucene-dev-return-290-qmlist-jakarta-lucene-dev-archive=jakarta.apache.org@jakarta.apache.org>

Delivered-To: apmail-jakarta-lucene-dev-archive@apache.org

Received: (qmail 87626 invoked from network); 29 Oct 2001 18:09:32 -0000

Received: from unknown (HELO nagoya.betaversion.org) (192.18.49.131)

by daedalus.apache.org with SMTP; 29 Oct 2001 18:09:32 -0000

Received: (qmail 26479 invoked by uid 97); 29 Oct 2001 18:09:09 -0000

Delivered-To: qmlist-jakarta-lucene-dev-archive@jakarta.apache.org

Received: (qmail 26414 invoked by uid 97); 29 Oct 2001 18:09:07 -0000

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

Precedence: bulk

List-Unsubscribe: <mailto:lucene-dev-unsubscribe@jakarta.apache.org>

List-Subscribe: <mailto:lucene-dev-subscribe@jakarta.apache.org>

List-Help: <mailto:lucene-dev-help@jakarta.apache.org>

List-Post: <mailto:lucene-dev@jakarta.apache.org>

List-Id: "Lucene Developers List" <lucene-dev.jakarta.apache.org>

Reply-To: "Lucene Developers List" <lucene-dev@jakarta.apache.org>

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

Received: (qmail 26373 invoked from network); 29 Oct 2001 18:09:07 -0000

Message-ID: <4BC270C6AB8AD411AD0B00B0D0493DF0EE7CBE@mail.grandcentral.com>

From: Doug Cutting <DCutting@grandcentral.com>

To: "'lucene-dev@jakarta.apache.org'" <lucene-dev@jakarta.apache.org>

Subject: RE: Performance in TermInfosReader

Date: Mon, 29 Oct 2001 09:58:00 -0800

MIME-Version: 1.0

X-Mailer: Internet Mail Service (5.5.2653.19)

Content-Type: text/plain;

charset="iso-8859-1"

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

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

> From: Dmitry Serebrennikov [mailto:dmitrys@earthlink.net]

>

> Doug, what would be an approach for making TermEnums "seekable" in an

> efficient manner?

I think you can make TermInfosReader more re-entrant by re-organizing the

code a bit. What you'd want to do is shift most of the fields and methods

from TermInfosReader into SegmentTermEnum, so that TermInfosReader becomes

simply a factory for SegmentTermEnums. Basically, all TermInfosReader needs

to do is cache a single SegmentTermEnum so that SegmentReader.get() requests

can be serviced without allocating a new TermEnum, and also be able to clone

this SegmentTermEnum to implement SegmentReader.terms(). The clones can all

share read-only copies of the contents of the .tii file (indexTerms,

indexInfos and indexPointers) and can thus all be independently seekable.

Note that the cached SegmentTermEnum holds the master copy of the .tis

InputStream, that, when closed, actually closes the file descriptor. The

.tii file is read entirely in the constructor, and so doesn't need to be

held open. Does this makes sense?

You could then add a public seek(Term) method to TermEnum. Seek(int) and

getPosition() would be very expensive to implement for SegmentsTermEnum, and

so probably shouldn't be made public. Or you could have them throw a

"please optimize your index" exception...

Doug

--

To unsubscribe, e-mail: <mailto:lucene-dev-unsubscribe@jakarta.apache.org>

For additional commands, e-mail: <mailto:lucene-dev-help@jakarta.apache.org>