http://mail-archives.apache.org/mod\_mbox/lucene-dev/200110.mbox/%3C4BC270C6AB8AD411AD0B00B0D0493DF0EE7C3B@mail.grandcentral.com%3E

From DCutting@grandcentral.com Mon Oct 8 16:33:49 2001

Return-Path: <DCutting@grandcentral.com>

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

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

Received: (qmail 29205 invoked from network); 8 Oct 2001 16:33:49 -0000

Received: from unknown (HELO mta.12.com) (65.198.8.41)

by daedalus.apache.org with SMTP; 8 Oct 2001 16:33:49 -0000

Received: (qmail 11138 invoked from network); 8 Oct 2001 16:31:11 -0000

Received: from unknown (HELO riker.grandcentral.com) (10.102.15.55)

by mta.12.com with SMTP; 8 Oct 2001 16:31:11 -0000

Received: by mail.grandcentral.com with Internet Mail Service (5.5.2653.19)

id <42Y1H2BC>; Mon, 8 Oct 2001 09:23:13 -0700

Message-ID: <4BC270C6AB8AD411AD0B00B0D0493DF0EE7C3B@mail.grandcentral.com>

From: Doug Cutting <DCutting@grandcentral.com>

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

Subject: RE: CachingDirectory contribution

Date: Mon, 8 Oct 2001 09:23:12 -0700

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

> From: Maik Schreiber [mailto:bZ@iq-computing.de]

>

> A while back I wrote a CachingDirectory implementation for

> Lucene which

> allows for caching an index on a local machine other than the "root"

> machine. This can be very useful for handling heavy load

> (such as David

> Snyder's 13 GB index :-))

>

> I'd really love to see it included in the Lucene package (I'm

> okay with

> it being put under the Apache license). If there's some

> interest I could

> provide the sources to the abstract CachingDirectory as well as actual

> FSCachingDirectory and RAMCachingDirectory implementations.

Please post them. If folks find them useful, they can be added to Lucene.

There are lots of places to cache. The operating system usually makes a

valiant effort to cache files on the local FS using available RAM. NFS

mounted files are also usually cached in RAM, and there are even NFS client

implementation which cache entire files on the local FS, so that if they

fall out of the RAM cache then a network read is not required. Explicitly

caching multi-gigabyte files in the Java heap is probably not practical.

But if NFS or the local FS do not cache things adequately, there might be a

place for Java-based caches that use either RAM or the local FS. I'm not

sure what they are, but that doesn't mean they don't exist! Benchmarking is

always the best way to determine these things, and if folks find that a

cache helps, let's add it.

Doug