<http://mail-archives.apache.org/mod_mbox/lucene-dev/200110.mbox/%3C4BC270C6AB8AD411AD0B00B0D0493DF0EE7C3C@mail.grandcentral.com%3E>

From DCutting@grandcentral.com Mon Oct 8 16:58:48 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 73817 invoked from network); 8 Oct 2001 16:58:48 -0000

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

by daedalus.apache.org with SMTP; 8 Oct 2001 16:58:48 -0000

Received: (qmail 12423 invoked from network); 8 Oct 2001 16:56:14 -0000

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

by mta.12.com with SMTP; 8 Oct 2001 16:56:14 -0000

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

id <42Y1H21C>; Mon, 8 Oct 2001 09:48:24 -0700

Message-ID: <4BC270C6AB8AD411AD0B00B0D0493DF0EE7C3C@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 09:48:18 -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: Dave Kor [mailto:davekkw@yahoo.com]

>

> This leads me to yet another of my buring questions..

> has anyone pushed Lucene to its limits yet? If so,

> what are they? What happens when Lucene hit its limit?

> Does it throw an exception? coredump?

There are many limits that could be hit. Lucene's design is that hard

limits should be hard to hit. Lucene only caches a few critical data

structures in memory, in order to keep from hitting the JVM's heap size

limit, relying instead on the file system's caches for performance. Lucene

uses 63-bit file pointers, so it will be a long time before raw index size

is a limit, however filesystems that do not support files larger than, e.g.,

2GB will limit things. Document and term numbers are 31-bit, so two billion

documents or terms is another limit that will will probably not be hit too

soon.

Performance for large indices is frequently governed by i/o performance. If

an index is larger than RAM then searches will need to read data from disk.

This can quickly become a bottleneck. A search for a term that occurs in a

million documents can require over 1MB of data, which can take some time to

read. With multiple searching threads, the disk can easily become a

bottleneck. Disk arrays can alleviate this, more RAM helps even more!

For some folks, queries that take over a second are unacceptable, for

others, ten seconds is okay.

Performance should be more-or-less linear: a two-million document index will

be almost twice as slow to search as a one-million document index. There

are lots of factors, including document size, CPU-speed, RAM-size, i/o

subsystem, but a rough rule-of-thumb for Lucene performance might be that,

in a "typical" configuration, it can search a million documents per second.

So if you need to search 20 million 100kB documents on a 100Mhz 386 with 8MB

of RAM with sub-second response time, Lucene will probably fail. But if you

need to search two million 2kB documents on a 500Mhz Pentium with 128MB of

RAM in a couple of seconds per query, you're probably okay.

Doug