

Re: FFT Multiplication on Disk?

Source: <http://coding.derkeiler.com/Archive/General/comp.programming/2007-09/msg00296.html>

- *From:* user923005 <dcorbit@xxxxxxxx>
 - *Date:* Thu, 20 Sep 2007 18:45:15 -0700
-

On Sep 20, 11:17 am, mike3 <mike4...@xxxxxxxx> wrote:

On Sep 20, 9:17 am, c...@xxxxxxxx (Richard Harter) wrote:

On Wed, 19 Sep 2007 23:37:04 -0700, mike3 <mike4...@xxxxxxxx> wrote:

On Sep 19, 5:55 pm, Ben Pfaff <b...@xxxxxxxxxxxxxxxx> wrote:

mike3 <mike4...@xxxxxxxx> writes:

However the problem is that this involves three in-place matrix transpositions of a square matrix. Does anyone know of an efficient algorithm to do those?

Here's one paper on the topic:<http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/8931/28260/0126...>

--
"J'avais trouv'e ma religion :
rien ne me parut plus important qu'un livre.
La biblioth`eque, j'y voyais un temple."
--Jean-Paul Sartre

Re: FFT Multiplication on Disk?

That looks like it's going to cost me a good sum of MONEY to get that paper. I don't have lots of that nice green stuff on hand given my current financial situation, unfortunately.

A simple alternative that you can roll on your own is to recursively partition the transposition. Here is the idea: Suppose we have a matrix that looks like this:

$$\begin{matrix} A & B \\ M & = & C & D \end{matrix}$$

where M is $2N \times 2N$ and A, B, C, and D are all $N \times N$. Transpose each component separately and then swap A and B. If you do things soothly the number of cache misses drops sharply.

Richard Harter,
c...@xxxxxxxxxxxxx://home.tiac.net/~cri,<http://www.varinoma.com>
But the rhetoric of holistic harmony can generate into a kind of dotty, Prince Charles-style mysticism. — Richard Dawkins

The problem though is that it is very non-sequential. Is this as close to sequential as one can get it, or is there better? I want to minimize the amount of disk operations, especially seeks, as much as possible. It's the `_disk_` part that's the real chokepoint here.—

Why are you multiplying numbers from disk anyway?

Buy more RAM. Even if you are multiplying numbers that consume over a gigabyte, you will be glad you did it that way. RAM is going to be thousands of times faster than disk. Here's an ad blurb for something that claims to be the fastest hard disk in the world:

"The Maxtor Atlas 15K SCSI drive is the fastest hard drive in the world. Its 3.2ms seek time enables 45% more I/Os per second than 10K RPM drives can achieve. The Atlas 15K drive can sustain up to 75MB/sec data transfer rate and is ideal for use in high-performance workstations, NAS and SAN environments, OLTP applications, enterprise servers and data mining... More applications. The drive is equipped with Maxtor-developed Ultra320 SCSI and is backwards compatible with all prior versions of SCSI. The Ultra320 interface includes MaxAdapt,

Re: FFT Multiplication on Disk?

a closed-loop method of improving signal quality by amplifying the fundamental frequency of the signal in the receiver while filtering noise and other undesirable components. MaxAdapt allows the drive to adapt to changing system conditions and components, which translates into lower error rates, easier integration, and increased bus efficiency for optimal system performance."

How many memory fetches do you think you can do in 3.2 ms (that's milliseconds, not microseconds)? Fast RAM can be accessed in 9 nanoseconds. That's about 355,556 times faster. Let's suppose that in RAM you could solve it in one second. On disk it would take 4 days.

Here is an example of memory pricing:

http://www.computermemoryoutlet.com/Compaq-ProLiant-8500_6%5Eslsh700-memory.htm

I would rather spend a little money and wait one second than spend nothing and wait 4 days.

.