Subject: Re: Help, no improvement in FFT speed on a multiprocessor system
Posted by Kenneth P. Bowman on Mon, 07 Sep 2009 13:40:36 GMT
View Forum Message <> Reply to Message

In article <4aa34391$1@darkstar>, "Marco" <null@null.net> wrote:

> I'm running IDL 7.1 on a Linux 2.6. This is an HP quad processor with each
> processors having 6 cores for 24 cores total.
>
> Doing large 2-D FFTs (>8Kx8K) I get no benefit from the extra processors.
> I can vary the number used in IDL from 24 down to 1, and see that the
> number actually of processors actually showing a load is the correct number,
> but from 4 to 24 threads, the speed is the same and no faster than Matlab,
> which uses only a single processor out of the 24.
>
> I've tried varied IDL_CPU_TPOOL_NTHREADS and IDL_CPU_TPOOL_MIN_ELTS, but
> have not been able to improve the results.
>
> Any suggestions?
>
> Thanks in advance,

Do your array dimensions have small prime factors (2, 3, 4, or 5)?

Ken Bowman

Subject: Re: Help, no improvement in FFT speed on a multiprocessor system
Posted by Marco on Mon, 07 Sep 2009 19:58:27 GMT
View Forum Message <> Reply to Message

The arrays are Nx8192 on a side with N a power of 2.

I increased N until the speed dropped of a cliff. Presumably cache/memory
thrashing.

"Kenneth P. Bowman" <k-bowman@null.edu> wrote in message
news:k-bowman-5D1FE7.08403607092009@news.tamu.edu...
> In article <4aa34391$1@darkstar>, "Marco" <null@null.net> wrote:
>
>> I'm running IDL 7.1 on a Linux 2.6. This is an HP quad processor with
>> each
>> processors having 6 cores for 24 cores total.
>>
>> Doing large 2-D FFTs (>8Kx8K) I get no benefit from the extra
>> processors.
>> I can vary the number used in IDL from 24 down to 1, and see that the

>> number actually of processors actually showing a load is the correct
>> number,
>> but from 4 to 24 threads, the speed is the same and no faster than
>> Matlab,
>> which uses only a single processor out of the 24.
>>
>> I've tried varied IDL_CPU_TPOOL_NTHREADS and IDL_CPU_TPOOL_MIN_ELTS, but
>> have not been able to improve the results.
>>
>> Any suggestions?
>>
>> Thanks in advance,
>
> Do your array dimensions have small prime factors (2, 3, 4, or 5)?
>
> Ken Bowman

Subject: Re: Help, no improvement in FFT speed on a multiprocessor system
Posted by Kenneth P. Bowman on Mon, 07 Sep 2009 21:30:36 GMT
View Forum Message <> Reply to Message

In article <4aa56605$1@darkstar>, "Marco" <null@null.net> wrote:

> The arrays are Nx8192 on a side with N a power of 2.
>
> I increased N until the speed dropped of a cliff. Presumably cache/memory
> thrashing.
>
>
> "Kenneth P. Bowman" <k-bowman@null.edu> wrote in message
> news:k-bowman-5D1FE7.08403607092009@news.tamu.edu...
>> In article <4aa34391$1@darkstar>, "Marco" <null@null.net> wrote:
>>
>>> I'm running IDL 7.1 on a Linux 2.6. This is an HP quad processor with
>>> each
>>> processors having 6 cores for 24 cores total.
>>>
>>> Doing large 2-D FFTs (>8Kx8K) I get no benefit from the extra
>>> processors.

I don't know how the Intel cache architecture works, but on some
processors (e.g., IBM Power), a cache miss causes a whole cache
line to be loaded from memory. If you are working on large arrays
and taking large strides through memory, every memory access
can cause a cache miss. This has the effect of completely
destroying the advantages of having a cache. Arrays dimensioned
by powers of two can be the worst cases.

I don't know an easy solution. You could do N 1-D FFTs of size 8192, transpose the output, and then do 8192 1-D FFTs of size N. That is, "manually" make a 2-D FFT by looping over the second dimension. It might possibly be faster than doing a 2-D FFT with miserable cache performance.

Ken Bowman
