Subject: IDL and Dual Processor PC's
Posted by Tanya Lancaster on Thu, 03 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

We are looking into purchasing a dual processor pc. I was wondering if there would be a notable increase in speed for running IDL programs. Right now we handle large medical image data sets and operations on the data sets take up to several hours on a PII -400, 256 mb memory.

-Tanya Lancaster

Subject: Re: IDL and Dual Processor PC's
Posted by Laurent Chardon on Fri, 04 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Mark Rivers wrote in message ...
<snip>

> A dual-processor system, running 2 instances of
> IDL simultaneously may be better value for money than a second complete
> computer. The incremental cost of the second CPU is not that high. One
> advantage is that you can put twice as much memory in this machine, and
have
> all of it available to a single IDL process when you need it

I agree. Not to mention that you can run as many instances of IDL with one license on one machine, but running IDL simultaneously on different machines would require one license per machine...

Laurent Chardon

Subject: Re: IDL and Dual Processor PC's
Posted by steinhh on Fri, 04 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

In article <FCsyCq.22s@midway.uchicago.edu>
rivers@cars3.uchicago.edu (Mark Rivers) writes:

> In article <7j8714$631$1@readme.uio.no>, steinhh@ulrik.uio.no
>> (Stein Vidar Hagfors Haugan) writes:
[..]
>> ..then you could be getting a lot more value for money if you
>> buy two single-processor systems, running them in parallell
>> "by hand"...

> I might disagree with this. A dual-processor system, running 2
> instances of IDL simultaneously may be better value for money than a
> second complete computer. The incremental cost of the second CPU is
> not that high. One advantage is that you can put twice as much
> memory in this machine, and have all of it available to a single IDL
> process when you need it. This is what we are doing - 1 GB of
> memory on a dual-processor 450 MHz Pentium, Windows NT. We are
> running up against the 32 bit memory limitations of Windows and IDL.
> Even with 3 GB of swap space, IDL can only access arrays just over 1
> GB. It doesn't take a very big 3-D array to reach that limit!

Quite true, Mark. It's really a question that requires some thought on how the system(s) are to be used, in addition to the actual prices per CPU/board/memory/etc.

I just wanted to raise the issue, because I've often had a lot of problems trying to explain to people the following scenario: You're going to do some complex calculation 1000 times over, and then later do statistics on the 1000 separate results. You want to think for a while before running a process 1000 times using 10 processors in parallell, rather than running a process 100 times on each of the processors individually. Which way is faster depends a *lot* on the problem at hand and on the machine architecture. With some problems and some (memory-cached) architectures, splitting up the problem on 10 processors can mean you're done maybe 20 times faster! With some problems, you might be unable to do it on a single machine anyway (memory limitations). But for a wide range of applications, you're getting the 1000 results faster by using good old sequential processing.

And of course, if you're waiting for a single enhanced medical image before you can start some life-saving operation after a car crash, you'd want to go with the parallell version even if it doesn't give you the most FLOPS/$.

Regards,

Stein Vidar

Subject: Re: IDL and Dual Processor PC's
Posted by rivers on Fri, 04 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

In article <7j8714$631$1@readme.uio.no>, steinhh@ulrik.uio.no
> (Stein Vidar Hagfors Haugan) writes:
>

> read_image,name2,huge_image
> fft_of_huge_image = fft(huge_image,-1)
> ;; .... processing....
> processed_image = fft(fft_of_huge_image,1)
> save_huge_image,processed_image
>
> ..then you could be getting a lot more value for money if you
> buy two single-processor systems, running them in parallell
> "by hand"...

I might disagree with this. A dual-processor system, running 2 instances of IDL simultaneously may be better value for money than a second complete computer. The incremental cost of the second CPU is not that high. One advantage is that you can put twice as much memory in this machine, and have all of it available to a single IDL process when you need it. This is what we are doing - 1 GB of memory on a dual-processor 450 MHz Pentium, Windows NT. We are running up against the 32 bit memory limitations of Windows and IDL. Even with 3 GB of swap space, IDL can only access arrays just over 1 GB. It doesn't take a very big 3-D array to reach that limit!

Mark Rivers

Subject: Re: IDL and Dual Processor PC's
Posted by steinhh on Fri, 04 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

In article <7j77fs$uo0$1@nnrp1.deja.com> Peter Mason <menakkis@my-deja.com> writes:
>
> "Tanya Lancaster" <lancaste@morph.ebme.cwru.edu> wrote:
>> We are looking into purchasing a dual processor pc. I was wondering
>> if there would be a notable increase in speed for running IDL
>> programs.
>> Right now we handle large medical image data sets and operations on
>> the data sets take up to several hours on a PII -400, 256 mb memory.
>
[...]
> Unless you are going to be writing multithreaded routines for accessing
> via IDL's call_external (or similar), or doing some sort of concurrent
> work like the Linux poster, I'd say that dual CPU is something of a
> luxury and the money's better spent on more memory or hardware RAID.
> (If you're routinely crunching through large datasets then it is likely
> that a good I/O subsystem will be your best investment by **F*A*R**.)

The answer to your question really depends on how you're processing the data. IDL is not going to do anything for you wrt taking advantage of dual processors, but if your processing involves just

a few, time-consuming (sequences of) operations, some of which are independent of the others, it's not *that* difficult to write e.g. CALL_EXTERNAL routines using e.g. RPC that'll pass data from one IDL process to another (slave) process, that does some processing while the master is doing (other) work on the data set.

Things to look out for is e.g.

```
fft_of_huge_image = fft(huge_image,-1)
fft_of_huge_image2 = fft(huge_image2,-1)
;; ....processing... involving both fft products
processed_image = fft(fft_of_huge_image,1)
processed_image2 = fft(fft_of_huge_image2,1)
```

If the fft's are taking a lot of your processing time, running a master/slave IDL process could be a good thing.

On the other hand, if your processing is more like this:

```
read_image,name,huge_image
fft_of_huge_image = fft(huge_image,-1)
;; .... processing....
processed_image = fft(fft_of_huge_image,1)
save_huge_image,processed_image

read_image,name2,huge_image
fft_of_huge_image = fft(huge_image,-1)
;; .... processing....
processed_image = fft(fft_of_huge_image,1)
save_huge_image,processed_image
```

..then you could be getting a lot more value for money if you buy two single-processor systems, running them in parallell "by hand"...

Regards,

Stein Vidar

Subject: Re: IDL and Dual Processor PC's
Posted by rivers on Fri, 04 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

In article <mgs-0306991006350001@teton.ivsoftware.com>, mgs@ivsoftware.com (Mike Schienle) writes:
> In article <7j62f0krm1@pale-rider.INS.CWRU.Edu>, "Tanya Lancaster"
> <lancaste@morph.ebme.cwru.edu> wrote:

>
>> We are looking into purchasing a dual processor pc. I was wondering if
>> there would be a notable increase in speed for running IDL programs.
>
> Nope. IDL is single-threaded, implying it cannot divvy it's processing up
> among multiple processors. Once you start IDL it stays on the same CPU.
> You can try some interesting things like spawning addition copies of IDL
> onto other processors. I tried that about 7 years ago on a Sun MP/670 with
> 4 processors. Not worth the effort was my conclusion.

It's not QUITE true that IDL is single-threaded. The IDLgrVolume object is
multi-threaded. Thus, volume rendering runs significantly faster on
multi-processor machines.

Mark Rivers (773) 702-2279 (office)
Argonne National Laboratory (630) 252-0422 (office)
Building 434A (630) 252-0405 (lab)
9700 South Cass Avenue (630) 252-1713 (beamline)
Argonne, IL 60439 (630) 252-0443 (FAX)

Subject: Re: IDL and Dual Processor PC's
Posted by Peter Mason on Fri, 04 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

"Tanya Lancaster" <lancaste@morph.ebme.cwru.edu> wrote:
> We are looking into purchasing a dual processor pc. I was wondering
> if there would be a notable increase in speed for running IDL
> programs.
> Right now we handle large medical image data sets and operations on
> the data sets take up to several hours on a PII -400, 256 mb memory.

We have a dual PII/300 running WinNT4. As has been noted by other
posters, the whole IDL environment is single-threaded and as such will
not run on more than 1 CPU at a time. However, the second CPU does
provide a varying amount of benefit - operating-system calls (such as
I/O) can involve it. At times you can see an overall CPU usage of up
to 70% or so while doing flat-out processing in ENVI, although this is
unusual and the typical figure is of course 50%.
Unless you are going to be writing multithreaded routines for accessing
via IDL's call_external (or similar), or doing some sort of concurrent
work like the Linux poster, I'd say that dual CPU is something of a
luxury and the money's better spent on more memory or hardware RAID.
(If you're routinely crunching through large datasets then it is likely
that a good I/O subsystem will be your best investment by **F*A*R**.)

Peter Mason

Sent via Deja.com http://www.deja.com/
Share what you know. Learn what you don't.

Subject: Re: IDL and Dual Processor PC's
Posted by mgs on Mon, 07 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

In article <woodford-0706991618590001@host32.pm4.laurel.us.net>,
woodford@essexcorp.com (Paul Woodford) wrote:

> IDL can now take advantage of multiple processors in rendering volume
> objects on WinNT, Irix, and Solaris - see the what's new section of the
> IDL manuals for more details. We obtained a 2x rendering speedup using
> this feature on a dual-processor NT machine that we tested. Volume
> objects take a long time to render, so this is a nice feature.
>
> I hope that RSI spreads this feature to other operating systems (such as
> Linux) and other operations where it should be easy to implement (such as
> FFT's).

Here's what I wrote earlier:
> Nope. IDL is single-threaded, implying it cannot divvy it's processing up
> among multiple processors. Once you start IDL it stays on the same CPU.
...
> I would love to hear that I'm wrong about this.

I couldn't be more happy to be wrong about some of my earlier statements.
Thanks, RSI :-)

--
Mike Schienle Interactive Visuals, Inc.
mgs@ivsoftware.com Remote Sensing and Image Processing
http://www.ivsoftware.com/ Analysis and Application Development

---

Subject: Re: IDL and Dual Processor PC's
Posted by woodford on Mon, 07 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

IDL can now take advantage of multiple processors in rendering volume
objects on WinNT, Irix, and Solaris - see the what's new section of the
IDL manuals for more details. We obtained a 2x rendering speedup using
this feature on a dual-processor NT machine that we tested. Volume
objects take a long time to render, so this is a nice feature.

I hope that RSI spreads this feature to other operating systems (such as Linux) and other operations where it should be easy to implement (such as FFT's).

--
Paul Woodford
woodford@essexcorp.com

Subject: Re: IDL and Dual Processor PC's
Posted by David Foster on Thu, 10 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Bruce L. Gotwols wrote:
> I am currently searching for a way to tell IDL which processor to
> use and any other tuning parameters which would make it more
> consistent.
>

I believe at least under Solaris you can use priocntl to "assign" a process to a CPU. Note that though you can bind a process to a CPU, you cannot bind a CPU to a process (make it exclusive).

Dave Foster
--

~~~~~~ ~~~~~~
David S. Foster Univ. of California, San Diego
Programmer/Analyst Brain Image Analysis Laboratory
foster@bial1.ucsd.edu Department of Psychiatry
(619) 622-5892 8950 Via La Jolla Drive, Suite 2240
La Jolla, CA 92037
~~~~~~ ~~~~~~
