Subject: Re: load sharing on multiple machines? Posted by Mark Rivers on Thu, 17 Jan 2002 04:07:44 GMT

View Forum Message <> Reply to Message

Mike Miller <mmiller3@iupui.edu> wrote in message news:yaghepmncy4.fsf@iupui.edu...

> Dear IDL'ers.

>

- > I'm a new IDL user and I've got some image reconstruction tasks
- > that are done on a slice-by-slice basis. I think that this will
- > make it simple for me to run IDL programs on several different
- > machines, each doing a subset of the slices, so that overall
- > processing time will be reduced. I've got a stack of Linux PCs
- > and some shiny new windows 2000 PCs to do this on. These windows
- > machines run at 1.8 GHz and I'd like to take advantage of all the
- > wasted cycles that aren't needed when the users are doing their
- > email and text processing.

>

- > Does anyone out there have any experience or suggestions on ways
- > to implement this? I can work it out with scripts on the linux
- > boxes, but I'm at a loss for how to do it with the windows
- > machines. Is it even possible to work them into the mix?
- > (without turning them into Linux boxes that is...)

I am doing tomography reconstruction that sounds similar to what you want to do. I am currently running on a single Windows machine, but with dual-CPUs. Both IDL and the Intel Math Kernal Library that I use for FFT-based reconstruction take advantage of both CPUs.

I have thought about what you are trying to do, use multiple machines, but have not done it yet.

Here are some possible approaches:

- One IDL process on one machine is the "master" and sends jobs to be run on other "server" machines. This could be accomplished by writing a file on the server, which IDL on that server is waiting to read. It could also be accomplished by running a server task each the server machine that listens for socket connections from the IDL master. IDL has a socket client interface.
- You could use MPI to parallelize the reconstruction. There is a group at Argonne who have this running on Windows machines for tomography reconstruction. Contact Francesco De Carlo (decarlo@aps.anl.gov) for more information. This is not using IDL, but I think IDL could be interfaced to it.

You might be interested in my Web page: http://cars.uchicago.edu/software/tomography.html This describes the IDL software that I've written for tomography reconstruction, including a new GUI interface to control it.

Mark Rivers

Subject: Re: load sharing on multiple machines?

Posted by Pavel A. Romashkin on Thu, 17 Jan 2002 17:36:15 GMT

View Forum Message <> Reply to Message

I admire your courage - starting in a new language right off with multithreading and clustering.

Sorry, I wish I could be of more help. But I am not ambitious enough to be useful :-(

Do a Google search though. There were a few threads on this, some even with solutions, in 2000.

Good luck,

Pavel

Mike Miller wrote:

>

- > I'm a new IDL user and I've got some image reconstruction tasks
- > that are done on a slice-by-slice basis. I think that this will
- > make it simple for me to run IDL programs on several different
- > machines...t

Subject: Re: load sharing on multiple machines? Posted by mmiller3 on Thu, 17 Jan 2002 19:46:53 GMT

View Forum Message <> Reply to Message

Thanks for the feedback, Mark. Our main systems also have dual-CPU slots, although we haven't yet filled the second one. Do you have any ball-park numbers for how much IDL performance improvement you see when you use two instead of one processors?

Mike

Subject: Re: load sharing on multiple machines?
Posted by Mark Rivers on Fri, 18 Jan 2002 04:34:19 GMT
View Forum Message <> Reply to Message

Michael A. Miller <mmiller3@iupui.edu> wrote in message news:87sn9469qq.fsf@lumen.med.iupui.edu...

- > Thanks for the feedback, Mark. Our main systems also have
- > dual-CPU slots, although we haven't yet filled the second one.
- > Do you have any ball-park numbers for how much IDL performance

> improvement you see when you use two instead of one processors?

I just ran a test on Linux IDL> a = findgen(1000,1000) IDL> b = findgen(1000,1000)+1. IDL> for i=0,1000 do c=a/b

I monitored the system performance with "top". Both CPUs went from 1% busy to 100% busy, indicating that IDL is using both CPUs effectively.

On the other hand:

IDL> for i=0,100 do c=fft(a,1)

the sum of the 2 CPUs busy was about 120%, indicating that it's not using both CPUs very well.

My observations on Windows are consistent with this, some operations fully utilize both CPUs, some don't.

Mark

Subject: Re: load sharing on multiple machines?
Posted by Craig Markwardt on Fri, 18 Jan 2002 15:33:37 GMT
View Forum Message <> Reply to Message

"Mark Rivers" <rivers@cars.uchicago.edu> writes:

- > Michael A. Miller <mmiller3@iupui.edu> wrote in message
- > news:87sn9469qq.fsf@lumen.med.iupui.edu...
- >> Thanks for the feedback, Mark. Our main systems also have
- >> dual-CPU slots, although we haven't yet filled the second one.
- >> Do you have any ball-park numbers for how much IDL performance
- >> improvement you see when you use two instead of one processors?
- >
- > I just ran a test on Linux
- > IDL> a = findgen(1000,1000)
- > IDL> b = findgen(1000,1000)+1.
- > IDL> for i=0,1000 do c=a/b

>

>

- > I monitored the system performance with "top". Both CPUs went from 1% busy
- > to 100% busy, indicating that IDL is using both CPUs effectively.
- > On the other hand:
- > IDL> for i=0,100 do c=fft(a,1)
- > the sum of the 2 CPUs busy was about 120%, indicating that it's not using
- > both CPUs very well.

I recall reading the What's New document for IDL 5.5, which says that

IDL 5.5.		
Craig		
Craig B. Markwardt, Ph.D. EMAIL:	. ,	

some mathematical operations can become multi-threaded, and hence use

multiple CPUs. I suspect that this is much more difficult to implement for the FFT, so it isn't. Of course, this only applies to