Subject: Re: Parallel Processing
Posted by Russell Ryan on Thu, 28 Jun 2012 14:27:47 GMT
View Forum Message <> Reply to Message

On Thursday, June 28, 2012 10:05:24 AM UTC-4, stefan....@gmail.com wrote:
> Hi
>
> I have developed a code which takes a couple of hours to run and I am aware of the fact that IDL automatically parallelizes some vector operations and one should prefer those instead of looping through arrays.
>
> I have done all that but still I know I could speed up things by a factor of 2 when I do certain things on 2 cores.
>
> For instance, somewhere in the program I pass some arrays to a function and this function then returns and equally large array with some calculated values. This is all done with one core since the operations in the function are not parallelized.
>
> However, I could split up the input arrays into to equally large parts and perform the calcualtions for each of those two on one core. In the end, when both are finished I could just concatenate the result-arrays.
>
> Is this possible in some easy way?
>
> thanks for your help :)

Hi Stefan,
I know of at least three ways to skin this cat, and they depend on the details of your problem and how much money you're willing to spend.

(1) Simply break the problem into several smaller bits, and run each bit in different IDL sessions. The OS will naturally divvy the computations accordingly. Of course, this is the easiest, but may not be practical for your problem.

(2) Employ the IDL_IDLBridge architecture built by ITT for IDL. This adds a new level of programming, but it's a way of running multiple IDL commands simultaneously. At first glance, this sounds tedious --- but I"ve done it 1000s of times and it's not too bad. This works best for problems where you want to preform some long task on one set of data and then go to a next set --- and the sets are unrelated. For example, have 100 objects and each object must be processed identically.

(3) There is some aftermarket software (I believe it's called FastDL) that you can build to do this. I'm not to familiar with it, but you can google it.

Russell