Subject: Re: When does IDL use MPs?
Posted by JD Smith on Fri, 15 Nov 2002 22:20:08 GMT
View Forum Message <> Reply to Message

On Wed, 13 Nov 2002 13:02:45 -0700, David Fanning wrote:

> Jonathan Greenberg (greenberg@ucdavis.edu) writes:
>
>> Do I need to specifically invoke multiprocessor calls in an IDL
>> procedure, or will it use MPs whenever appropriate?
>
> Oh, oh. Those marketing guys have gotten out again. :-(
>
> I'm afraid IDL is not multi-threaded. It can use multiprocessors in one
> very narrowly defined situation: when it is rendering object graphics
> volumes. Other than that, you can invoke until the cows come home, but
> it's not going to help much. Prayer would be more effective, I think.
> :-)
>

Well, it's slightly better than that. IDL is designed to use its thread pool for a number of the most important operations and routines (WHERE, FFT, basic array arithmetic, etc.) on MP machines whenever appropriate. The bad news is that the overhead for spawning a separate thread to compute, say, the total of a big array is significant, motivating the default choice for the minimum number of elements before threading occurs to be large:

```
help,!CPU,/STRUCTURES
** Structure !CPU, 6 tags, length=24, data length=24:
   HW_VECTOR       LONG              0
   VECTOR_ENABLE   LONG              0
   HW_NCPU         LONG              2
   TPOOL_NTHREADS  LONG              2
   TPOOL_MIN_ELTS  LONG         100000
   TPOOL_MAX_ELTS  LONG              0
```

I.e. unless you're operating on an array with more than 100,000 elements, you are out of luck (TPOOOL_MIN_ELTS). You can change this value globally with the CPU procedure, or hand tune it in the call to any of the 50 or so routines which are threaded, but 100,000 is usually about right.

When the magic is right, it really cooks on my dual-processor system:

```
IDL> r=randomu(sd,10000000L)
IDL> t=systime(1) & tot=total(r,TPOOL_MIN_ELTS=10000001L) & print,systime(1)-t
      0.11507905
```

```
IDL> t=systime(1) & tot=total(r,TPOOL_MIN_ELTS=10000000L) & print,systime(1)-t
      0.061390042
```

I find measurable speedups even for 50,000 elements, but it really shines on much larger chunks of data. It also depends on the function, of course. Experiment and see what you find.

JD
