Subject: Re: Q: IDL benchmarks
Posted by Ken Knighton on Sat, 24 Feb 1996 08:00:00 GMT
View Forum Message <> Reply to Message

rivers@cars3.uchicago.edu (Mark Rivers) wrote:
> In article <4gkdde$9nm@reznor.larc.nasa.gov>, zawodny@arbd0.larc.nasa.gov (Joseph M Zawodny) writes:
>> In article <sterner.825034453@warble.jhuapl.edu> sterner@warble.jhuapl.edu (Ray Sterner) writes:
>>> Bringfried Stecklum <stecklum@gwaihir.astro.uni-jena.de> writes:
>>>
>>>> Is there a collection of benchmark test results which shows how IDL behaves
>>>> on different platforms? There is no reference to this issue in the FAQ.
> Folks,
>
> In doing this compilation, please use TIME_TEST2, not TIME_TEST. TIME_TEST has
> a number of bugs, particularly for disk I/O on VMS. TIME_TEST2 also runs tests
> for longer so that the times are more meaningful.

Something else to keep in mind: Is the disk local or remote (nfs)? We use an NFS file server and when I changed to a local area, I reduced the disk i/o time by a factor of 100 (as one would expect).

HP-800 T500 w/ 3 CPUs, 512M memory (64M process size), HP-UX 9.04:

IDL> time_test2

1	1.55186	Empty For loop, 2000000 times
2	1.03341	Call empty procedure (1 param) 100,000 times
3	0.662755	Add 100,000 integer scalars and store
4	0.611896	25,000 scalar loops each of 5 ops, 2 =, 1 if)
5	0.615466	Mult 512 by 512 byte by constant and store, 10 times
6	0.486863	Shift 512 by 512 byte and store, 100 times
7	1.07607	Add constant to 512 x 512 byte array and store, 50 times
8	0.738948	Add two 512 by 512 byte images and store, 30 times
9	1.91268	Mult 512 by 512 floating by constant and store, 30 times
10	1.76016	Add constant to 512 x 512 floating and store, 30 times
11	2.78627	Add two 512 by 512 floating images and store, 30 times
12	0.531054	Generate 225000 random numbers
13	0.509610	Invert a 150 by 150 random matrix
14	0.496082	LU Decomposition of a 150 by 150 random matrix
15	0.740078	Transpose 256 x 256 byte, FOR loops
16	0.0969341	Transpose 256 x 256 byte, row and column ops
17	0.00765991	Transpose 256 x 256 byte, transpose function
18	1.57419	Log of 100,000 numbers, FOR loop
19	0.166916	Log of 100,000 numbers, vector ops
20	0.687387	131072 point forward plus inverse FFT
21	2.96173	Smooth 512 by 512 byte array, 5x5 boxcar, 10 times
22	0.252672	Smooth 512 by 512 floating array, 5x5 boxcar, 2 times

23 0.648114 Write and read 512 by 512 byte array x 20
21.9088=Total Time, 0.60454475=Geometric mean, 23 tests.

Using NFS disk, test 23 completed in 65 seconds. This skewed my geometric mean up to 0.72.

Ken Knighton knighton@gav.gat.com knighton@cts.com
General Atomics
San Diego, CA
