Subject: Re: Q: IDL benchmarks
Posted by szoonem on Fri, 23 Feb 1996 08:00:00 GMT
View Forum Message <> Reply to Message

In article <4gkdde$9nm@reznor.larc.nasa.gov>, zawodny@arbd0.larc.nasa.gov (Joseph M Zawodny) wrote:

>
> I agree that the individual test times should be kept.
>
> I have a file of a few systems that I have run TIME_TEST on, ...
> ...

For those who are interested, here are the numbers for a PowerMac 7100/80 with 40MB of RAM running IDL 4.0. Virtual memory is turned off.

```
      1     0.716667 Empty For loop, 1 million times
      2      1.63333 Call empty procedure (1 param) 100,000 times
      3     0.666667 Add 100,000 integer scalars and store
      4     0.766667 25,000 scalar loops each of 5 ops, 2 =, 1 if)
      5     0.316667 Mult 512 by 512 byte by constant and store, 10 times
      6     0.300000 Shift 512 by 512 byte and store, 10 times
      7     0.300000 Add constant to 512 x 512 byte array and store, 10 times
      8     0.300000 Add two 512 by 512 byte images and store, 10 times
      9     0.466667 Mult 512 by 512 floating by constant and store, 10 times
     10      1.10000 Add constant to 512 x 512 floating and store, 10 times
     11     0.966667 Add two 512 by 512 floating images and store, 10 times
     12     0.183333 Invert a 100 by 100 random matrix
     13     0.766667 Transpose 256 x 256 byte, FOR loops
     14     0.200000 Transpose 256 x 256 byte, row and column ops
     15    0.0666666 Transpose 256 x 256 byte, transpose function
     16      1.76667 Log of 100,000 numbers, FOR loop
     17     0.350000 Log of 100,000 numbers, vector ops
     18      1.70000 Add two 100000 element floating vectors, FOR loop
     19    0.0666667 Add two 100000 element floating vectors, vector op
     20     0.350000 65536 point real to complex FFT
     21     0.216667 Smooth 512 by 512 byte array, 5x5 boxcar
     22     0.200000 Smooth 512 by 512 floating array, 5x5 boxcar
     23      5.30000 Write and read 10 512 by 512 byte arrays
    18.7000=Total Time,       0.46858267=Geometric mean,      23 tests.
```

I feel I have to add a comment here. Even though I am very happy with running IDL on my Mac, it has some drawbacks that could be deadly. For instance, we have to turn virtual memory on if we need to deal with very large arrays. This makes every thing extremely slow.

---------- -----
| Saeid Zoonematkermani | E-Mail: szoonem@astro.sunysb.edu |

| Earth and Space Sciences | Voice: (516)632-8237 |
| State University of New York | Fax: (516)632-8742 |
| Stony Brook, NY 11794-2100 | |
| Home Page --> http://ozone.ess.sunysb.edu/ |

Subject: Re: Q: IDL benchmarks
Posted by zawodny on Fri, 23 Feb 1996 08:00:00 GMT
View Forum Message <> Reply to Message

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.
>
> I would be happy to add to the FAQ a reference to the URL of such a
> collection if one exists. If somebody decides to put one together
> it would be very useful to include enough information about the
> systems to be able to interpret the results. Just what should be
> included might be worth a bit of discussion here before going to the
> trouble of collecting the timing results. For example, I wonder just
> how much meaning time test number 23 (Write and read 10 512 by 512
> byte arrays) really has. I would think that a big disk cache will give
> a lower result then expected for a much larger number of writes.
>
> Also it would be nice to include all the component times and not
> just the totals. However the totals are still a useful summary.
> Such a summary is already available in the online help (for V 4.01
> at least) under TIME_TEST. I am happy to see that my HP system beats
> the DEC Alpha with 224 Mb of memory (I only have 144 Mb).
> However a coworker's Pentium beats my HP. Time to upgrade I guess.

I agree that the individual test times should be kept.

I have a file of a few systems that I have run TIME_TEST on, it is enclosed below. Please note that the Gateway P5-60 had a elapsed time of 0.000000 for test 15 which of course ruins the value of the geometric mean. A value of 0.05 was used to calculate the geometric mean. The PCs typically have 16 or 32 MB of ram while the Alphas had 128 and 96MB respectively. I'll post the results from a Micron 200MHz Pentium Pro in a few weeks if there is interest.

Result IDL's TIME_TEST.

Test Gateway AMD AMD DEC Alpha DEC Alpha

| # | P5-60 | DX4-100 | DX4-133 | 3000/600 | 600/266 |
|---|---|---|---|---|---|
| 1 | 0.60 | 1.65 | 0.77 | 0.46 | 0.22 |
| 2 | 0.99 | 2.74 | 1.26 | 0.89 | 0.32 |
| 3 | 0.66 | 1.98 | 0.94 | 0.49 | 0.21 |
| 4 | 0.88 | 2.03 | 0.88 | 0.54 | 0.21 |
| 5 | 0.77 | 0.60 | 0.50 | 0.48 | 0.21 |
| 6 | 0.39 | 0.28 | 0.16 | 0.05 | 0.05 |
| 7 | 0.66 | 0.44 | 0.33 | 0.21 | 0.20 |
| 8 | 0.71 | 0.54 | 0.60 | 0.26 | 0.20 |
| 9 | 1.20 | 1.05 | 0.99 | 0.35 | 0.53 |
| 10 | 1.21 | 0.77 | 0.94 | 0.15 | 0.32 |
| 11 | 1.38 | 1.26 | 1.64 | 0.49 | 0.64 |
| 12 | 0.22 | 0.71 | 0.49 | 0.48 | 0.34 |
| 13 | 1.21 | 1.70 | 0.83 | 0.69 | 0.60 |
| 14 | 0.22 | 0.22 | 0.11 | 0.09 | 0.06 |
| 15 | 0.00! | 0.06 | 0.05 | 0.04 | 0.02 |
| 16 | 2.41 | 4.39 | 2.03 | 1.70 | 1.16 |
| 17 | 0.50 | 0.44 | 0.28 | 0.11 | 0.08 |
| 18 | 2.31 | 4.34 | 2.14 | 1.45 | 1.19 |
| 19 | 0.05 | 0.05 | 0.11 | 0.04 | 0.02 |
| 20 | 0.49 | 0.77 | 0.55 | 0.33 | 0.33 |
| 21 | 0.38 | 0.33 | 0.16 | 0.13 | 0.15 |
| 22 | 0.22 | 0.71 | 0.43 | 0.20 | 0.20 |
| 23 | 0.71 | 12.24 | 2.31 | 1.43 | 2.55 |
| | 18.17 | 39.30 | 18.50 | 11.07 | 9.80 Total Time |
| | 0.55! | 0.82 | 0.53 | 0.30 | 0.23 Geometric mean |

# Test
1 Empty For loop, 1 million times
2 Call empty procedure (1 param) 100,000 times
3 Add 100,000 integer scalars and store
4 25,000 scalar loops each of 5 ops, 2 =, 1 if)
5 Mult 512 by 512 byte by constant and store, 10 times
6 Shift 512 by 512 byte and store, 10 times
7 Add constant to 512 x 512 byte array and store, 10 times
8 Add two 512 by 512 byte images and store, 10 times
9 Mult 512 by 512 floating by constant and store, 10 times
10 Add constant to 512 x 512 floating and store, 10 times
11 Add two 512 by 512 floating images and store, 10 times
12 Invert a 100 by 100 random matrix
13 Transpose 256 x 256 byte, FOR loops
14 Transpose 256 x 256 byte, row and column ops
15 Transpose 256 x 256 byte, transpose function
16 Log of 100,000 numbers, FOR loop
17 Log of 100,000 numbers, vector ops
18 Add two 100000 element floating vectors, FOR loop

19 Add two 100000 element floating vectors, vector op
20 65536 point real to complex FFT
21 Smooth 512 by 512 byte array, 5x5 boxcar
22 Smooth 512 by 512 floating array, 5x5 boxcar
23 Write and read 10 512 by 512 byte arrays

--
Dr. Joseph M. Zawodny KO4LW NASA Langley Research Center
E-mail: J.M.Zawodny@LaRC.NASA.gov MS-475, Hampton VA, 23681-0001

---

Subject: Re: Q: IDL benchmarks
Posted by sterner on Fri, 23 Feb 1996 08:00:00 GMT
View Forum Message <> Reply to Message

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.

I would be happy to add to the FAQ a reference to the URL of such a collection if one exists. If somebody decides to put one together it would be very useful to include enough information about the systems to be able to interpret the results. Just what should be included might be worth a bit of discussion here before going to the trouble of collecting the timing results. For example, I wonder just how much meaning time test number 23 (Write and read 10 512 by 512 byte arrays) really has. I would think that a big disk cache will give a lower result then expected for a much larger number of writes.

Also it would be nice to include all the component times and not just the totals. However the totals are still a useful summary. Such a summary is already available in the online help (for V 4.01 at least) under TIME_TEST. I am happy to see that my HP system beats the DEC Alpha with 224 Mb of memory (I only have 144 Mb). However a coworker's Pentium beats my HP. Time to upgrade I guess.

Ray Sterner sterner@tesla.jhuapl.edu
The Johns Hopkins University North latitude 39.16 degrees.
Applied Physics Laboratory West longitude 76.90 degrees.
Laurel, MD 20723-6099
WWW Home page: http://fermi.jhuapl.edu/s1r/people/res/res.html

---

Subject: Re: Q: IDL benchmarks
Posted by Ken Knighton on Sat, 24 Feb 1996 08:00:00 GMT

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

---