Subject: Q: IDL benchmarks
Posted by Bringfried Stecklum on Wed, 21 Feb 1996 08:00:00 GMT
View Forum Message <> Reply to Message

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.

Thanks for any advice.

+-----------------------------------------------------+
| Bringfried Stecklum                                 |
| MPG research unit "Dust in Star Forming Regions"    |
| Schillergaesschen 3, FRG-07745 Jena, Germany        |
| Internet: stecklum@astro.uni-jena.de                |
| FAX: 049 3641 55594                                 |
| Phone: 049 3641 55593 or 049 3641 630307            |
+-----------------------------------------------------+

Subject: Re: Q: IDL benchmarks
Posted by Andy Hsia on Sun, 25 Feb 1996 08:00:00 GMT
View Forum Message <> Reply to Message

Our DEC Alpha 3000/600 with 48Mb RAM achieved the following:

| | | |
|---|---|---|
| 1 | 0.552992 | Empty For loop, 1 million times |
| 2 | 0.848720 | Call empty procedure (1 param) 100,000 times |
| 3 | 0.412848 | Add 100,000 integer scalars and store |
| 4 | 0.476864 | 25,000 scalar loops each of 5 ops, 2 =, 1 if) |
| 5 | 0.511424 | Mult 512 by 512 byte by constant and store, 10 times |
| 6 | 0.0644159 | Shift 512 by 512 byte and store, 10 times |
| 7 | 0.198128 | Add constant to 512 x 512 byte array and store, 10 times |
| 8 | 0.226032 | Add two 512 by 512 byte images and store, 10 times |
| 9 | 0.537776 | Mult 512 by 512 floating by constant and store, 10 times |
| 10 | 0.165920 | Add constant to 512 x 512 floating and store, 10 times |
| 11 | 0.393904 | Add two 512 by 512 floating images and store, 10 times |
| 12 | 0.237168 | Invert a 100 by 100 random matrix |
| 13 | 0.594960 | Transpose 256 x 256 byte, FOR loops |
| 14 | 0.0663681 | Transpose 256 x 256 byte, row and column ops |
| 15 | 0.00975990 | Transpose 256 x 256 byte, transpose function |
| 16 | 1.58853 | Log of 100,000 numbers, FOR loop |
| 17 | 0.0722239 | Log of 100,000 numbers, vector ops |
| 18 | 1.33242 | Add two 100000 element floating vectors, FOR loop |
| 19 | 0.0117120 | Add two 100000 element floating vectors, vector op |
| 20 | 0.173728 | 65536 point real to complex FFT |
| 21 | 0.213744 | Smooth 512 by 512 byte array, 5x5 boxcar |
| 22 | 0.119072 | Smooth 512 by 512 floating array, 5x5 boxcar |

23 1.80675 Write and read 10 512 by 512 byte arrays
10.6155 Total Time
0.24125159 Geometric mean, 23 tests.

---

Subject: Re: Q: IDL benchmarks
Posted by Mirko Vukovic on Mon, 26 Feb 1996 08:00:00 GMT
View Forum Message <> Reply to Message

Andy Hsia wrote:
>
> Our DEC Alpha 3000/600 with 48Mb RAM achieved the following:
> snip
> 0.119072 Smooth 512 by 512 floating array, 5x5 boxcar
> 23 1.80675 Write and read 10 512 by 512 byte arrays
> 10.6155 Total Time
> 0.24125159 Geometric mean, 23 tests.

What is really funny is that my P-120 (16 Meg RAM) at home gives 9.55 for total time, .305 for the geometric mean. Do these numbers make sense?
--
Mirko Vukovic, Ph.D. mirko.vukovic@grc.varian.com
Varian Research Center Phone: (415) 424-4969
3075 Hansen Way, M/S K-109 Fax: (415) 424-6988
Palo Alto, CA 94304-1025

---

Subject: Re: Q: IDL benchmarks
Posted by Robert Moss on Mon, 26 Feb 1996 08:00:00 GMT
View Forum Message <> Reply to Message

I just wanted to mention that before everyone runs out and buys PVWave on an SGI they should look carefully at these numbers: they are for much fewer iterations that the corresponding tests in IDL, so direct comparison to most of the other recent postings without some scaling is not valid.

Robert M. Moss, Ph.D.
Texaco Inc.
mossrm@texaco.com

Watson R J wrote:
>

> Running PV-WAVE CL Version 5.00 (sgi IRIX mipseb) on an SG Indy
> with 100MHz R4400 and 144Mb time_test gives this ...
>
>  1  0.156311 Empty For loop, 10,000 times
>  2  0.467805 Call empty procedure (1 param) 10000 times
>  3  0.142651 Add 10,000 integer scalars and store
>  4  0.241405 5000 scalar arithmetic loops
>  5  0.0721331 Mult 512 by 512 byte by constant and store
>  6  0.0324579 Shift 512 by 512 byte and store
>  7  0.0448910 Add a constant to a 512 by 512 byte array and store
>  8  0.0431681 Add two 512 by 512 byte images and store
>  9  0.318135 Mult 512 by 512 floating by constant and store
>  10  0.0612410 Shift 512 by 512 floating and store
>  11  0.0830290 Add a constant to a 512 by 512 floating array and store
>  12  0.326699 Add two 512 by 512 floating images and store
>  13  0.0674850 Invert a 50 by 50 random matrix
>  14  1.18785 Transpose 256 x 256 byte, FOR loops
>  15  0.133863 Transpose 256 x 256 byte, row and column ops
>  16  0.0236490 Transpose 256 x 256 byte, transpose
>  17  0.0209789 Log of 10000 numbers, vector ops
>  18  0.321045 Log of 10000 numbers, FOR loop
>  19  0.00391400 Add two 10000 element floating vectors, vector op
>  20  0.267292 Add two 10000 element floating vectors, FOR loop
>  21  0.0432030 8192 point real to complex FFT
>  22  0.372203 Smooth 512 by 512 byte array, 5x5 boxcar
>  23  0.256296 Smooth 512 by 512 floating array, 5x5 boxcar
>  24  0.186448 Write and read 10 512 by 512 byte arrays
>  4.87416 Total Time
>  0.10921606 Geometric mean,  24 tests.
>
> posted out of interest ...
> --
> ---------- ----------
> Robert J Watson.  "When in a hole... stop digging ..."
> University of Essex,  Electronic Systems Engineering Dept.
> Wivenhoe Park,  Internet: watsr@essex.ac.uk
> Colchester,  Tel : +44 1206 873029 (direct)
> CO4 3SQ.  Fax : +44 1206 872900
> ---------- ----------

********** **********
This is not necessarily the opinion of Texaco Inc.
********** **********

---

Subject: Re: Q: IDL benchmarks
Posted by watsr on Mon, 26 Feb 1996 08:00:00 GMT

Running PV-WAVE CL Version 5.00 (sgi IRIX mipseb) on an SG Indy
with 100MHz R4400 and 144Mb time_test gives this ...

```
 1     0.156311 Empty For loop, 10,000 times
 2     0.467805 Call empty procedure (1 param) 10000 times
 3     0.142651 Add 10,000 integer scalars and store
 4     0.241405 5000 scalar arithmetic loops
 5    0.0721331 Mult 512 by 512 byte by constant and store
 6    0.0324579 Shift 512 by 512 byte and store
 7    0.0448910 Add a constant to a 512 by 512 byte array and store
 8    0.0431681 Add two 512 by 512 byte images and store
 9     0.318135 Mult 512 by 512 floating by constant and store
10    0.0612410 Shift 512 by 512 floating and store
11    0.0830290 Add a constant to a 512 by 512 floating array and store
12     0.326699 Add two 512 by 512 floating images and store
13    0.0674850 Invert a 50 by 50 random matrix
14      1.18785 Transpose 256 x 256 byte, FOR loops
15     0.133863 Transpose 256 x 256 byte, row and column ops
16    0.0236490 Transpose 256 x 256 byte, transpose
17    0.0209789 Log of 10000 numbers, vector ops
18     0.321045 Log of 10000 numbers, FOR loop
19   0.00391400 Add two 10000 element floating vectors, vector op
20     0.267292 Add two 10000 element floating vectors, FOR loop
21    0.0432030 8192 point real to complex FFT
22     0.372203 Smooth 512 by 512 byte array, 5x5 boxcar
23     0.256296 Smooth 512 by 512 floating array, 5x5 boxcar
24     0.186448 Write and read 10 512 by 512 byte arrays
4.87416 Total Time
0.10921606 Geometric mean,      24 tests.
```

posted out of interest ...
--
---------- ----------
Robert J Watson.  "When in a hole... stop digging ..."
University of Essex,  Electronic Systems Engineering Dept.
Wivenhoe Park,  Internet:  watsr@essex.ac.uk
Colchester,  Tel  : +44 1206 873029 (direct)
CO4 3SQ.  Fax  : +44 1206 872900
---------- ----------

## Subject: Re: Q: IDL benchmarks
## Posted by Mark Baldwin on Tue, 27 Feb 1996 08:00:00 GMT



The following is from a Pentium 133 w/ SCSI disk, 32 Mb RAM, running IDL 4.01, PCI

bus. Could anyone post the result for a Pentium Pro 200?

NOTE: This is NOT the same time_test from PV_WAVE! E.G. test #1 for IDL is run 1 million times, while PV_WAVE runs it 10,000 times.

```
IDL> time_test
      1     0.431000 Empty For loop, 1 million times
      2     0.601000 Call empty procedure (1 param) 100,000 times
      3     0.461000 Add 100,000 integer scalars and store
      4     0.490000 25,000 scalar loops each of 5 ops, 2 =, 1 if)
      5     0.501000 Mult 512 by 512 byte by constant and store, 10 times
      6     0.240000 Shift 512 by 512 byte and store, 10 times
      7     0.381000 Add constant to 512 x 512 byte array and store, 10 times
      8     0.431000 Add two 512 by 512 byte images and store, 10 times
      9     0.691000 Mult 512 by 512 floating by constant and store, 10 times
     10      0.711000 Add constant to 512 x 512 floating and store, 10 times
     11      0.891000 Add two 512 by 512 floating images and store, 10 times
     12      0.210000 Invert a 100 by 100 random matrix
     13      0.641000 Transpose 256 x 256 byte, FOR loops
     14      0.100000 Transpose 256 x 256 byte, row and column ops
     15     0.0200000 Transpose 256 x 256 byte, transpose function
     16       1.80200 Log of 100,000 numbers, FOR loop
     17      0.321000 Log of 100,000 numbers, vector ops
     18       1.15100 Add two 100000 element floating vectors, FOR loop
     19     0.0500001 Add two 100000 element floating vectors, vector op
     20      0.320000 65536 point real to complex FFT
     21      0.260000 Smooth 512 by 512 byte array, 5x5 boxcar
     22      0.160000 Smooth 512 by 512 floating array, 5x5 boxcar
     23      0.430000 Write and read 10 512 by 512 byte arrays
     11.2940=Total Time,       0.34653037=Geometric mean,      23 tests.
```

Mark Baldwin
Northwest Research Associates
Bellevue, WA USA

---

## Subject: Re: Q: IDL benchmarks
Posted by Robert Cannon on Wed, 28 Feb 1996 08:00:00 GMT
View Forum Message <> Reply to Message

```
>>       10.6155 Total Time
>>       0.24125159 Geometric mean,      23 tests.
>
>  What is really funny is that my P-120 (16 Meg RAM) at home gives 9.55
>  for total time, .305 for the geometric mean. Do these numbers make
>  sense?
>  --
```

I get the same results between an alpha and a P-120, but for the
numerical stuff I  run most of the time the alpha is
three times faster - If you look at the "vector ops" tests
in the two cases you get about the same factor of three:

As far as I can see, the alpha is very good at vectorisable problems,
and no better than a P-120 for anything else.

Robert

---

Subject: Re: Q: IDL benchmarks
Posted by mahan on Thu, 29 Feb 1996 08:00:00 GMT
View Forum Message <> Reply to Message

To add to the benchmarks, here are the Results that I get on my notebook.

Toshiba 405CS:  75Mhz Pentium w/ 40MB RAM running win95 & IDL 4.0.1

For time_test I get:
   13.9400=Total Time,       0.47373837=Geometric mean,      23 tests.

and for time_test2 I get (all times shown):

```
IDL> time_test2
      1      1.10000
Empty For loop,     2000000 times
      2     0.930000
Call empty procedure (1 param) 100,000 times
      3     0.830000
Add 100,000 integer scalars and store
      4     0.660000
25,000 scalar loops each of 5 ops, 2 =, 1 if)
      5     0.650000
Mult 512 by 512 byte by constant and store, 10 times
      6      2.42000
Shift 512 by 512 byte and store, 100 times
      7      1.92000
Add constant to 512 x 512 byte array and store, 50 times
      8      1.21000
Add two 512 by 512 byte images and store, 30 times
      9      2.30000
Mult 512 by 512 floating by constant and store, 30 times
     10      2.20000
Add constant to 512 x 512 floating and store, 30 times
     11      2.69000
```

Add two 512 by 512 floating images and store, 30 times
12 0.330000
Generate 225000 random numbers
13 0.820000
Invert a 150 by 150 random matrix
14 0.220000
LU Decomposition of a 150 by 150 random matrix
15 0.770000
Transpose 256 x 256 byte, FOR loops
16 0.170000
Transpose 256 x 256 byte, row and column ops
17 0.0500000
Transpose 256 x 256 byte, transpose function
18 2.15000
Log of 100,000 numbers, FOR loop
19 0.490000
Log of 100,000 numbers, vector ops
20 2.75000
131072 point forward plus inverse FFT
21 3.13000
Smooth 512 by 512 byte array, 5x5 boxcar, 10 times
22 0.380000
Smooth 512 by 512 floating array, 5x5 boxcar, 2 times
23 0.710000
Write and read 512 by 512 byte array x 20
28.8800=Total Time,
0.85407678=Geometric mean, 23
tests.

_____

Stephen L. Mahan
mahan@aurora.phys.utk.edu