Subject: 2x speed boost from Mac OS X 10.5 to 10.6 upgrade
Posted by M. Katz on Mon, 28 Sep 2009 03:55:14 GMT
View Forum Message <> Reply to Message

I just upgraded a late-2007 iMac from OS X 10.5 to 10.6, and there's a 2x speed increase in IDL 7.1! I thought I had seen the same thing with my MacBookPro, but I wasn't sure. So I ran IDL's time testing routines before and after the upgrade--about 1 hour apart.

Short Answer from TIME_TEST2
10.5: 0.376995 seconds
10.6: 0.198143 seconds

GRAPHICS_TIMES: 0.132045 seconds, no difference

Here are the long results from TIME_TEST2. Same exact computer. All other applications were closed and quit. Both tests were run 10 times in a row so the minimum time could be selected.

In both cases:
Shading language version: 1.20
Hardware shaders are available

========== Mac OS X 10.5 ==========
IDL> time_test2
|TIME_TEST2 performance for IDL 7.1:
| OS_FAMILY=unix, OS=darwin, ARCH=x86_64
| Sun Sep 27 18:40:33 2009
1 0.0209451 Empty For loop, 2000000 times
2 0.0127928 Call empty procedure (1 param) 100,000 times
3 0.00810790 Add 100,000 integer scalars and store
4 0.00997496 25,000 scalar loops each of 5 ops, 2 =, 1 if)
5 0.00430989 Mult 512 by 512 byte by constant and store, 10 times
6 0.0243580 Shift 512 by 512 byte and store, 100 times
7 0.0179489 Add constant to 512 x 512 byte array and store, 50 times
8 0.0116029 Add two 512 by 512 byte images and store, 30 times
9 0.0418870 Mult 512 by 512 floating by constant and store, 30 times
10 0.0328600 Add constant to 512 x 512 floating and store, 40 times
11 0.0520871 Add two 512 by 512 floating images and store, 30 times
12 0.00435996 Generate 225000 random numbers
13 0.00518799 Invert a 150 by 150 random matrix

```
      14   0.00181079 LU Decomposition of a 150 by 150 random matrix
      15   0.00776792 Transpose 256 x 256 byte, FOR loops
      16    0.0100720 Transpose 256 x 256 byte, row and column ops x
10
      17   0.00322008 Transpose 256 x 256 byte, TRANSPOSE function x
10
      18    0.0181770 Log of 100,000 numbers, FOR loop
      19   0.00187993 Log of 100,000 numbers, vector ops
      20    0.0172210       131072 point forward plus inverse FFT
      21    0.0393300 Smooth 512 by 512 byte array, 5x5 boxcar, 10
times
      22   0.00582004 Smooth 512 by 512 floating array, 5x5 boxcar, 2
times
      23    0.0252740 Write and read 512 by 512 byte array x 20
    0.376995=Total Time,      0.011138194=Geometric mean,      23
tests.

========== Mac OS X 10.6 ==========
|TIME_TEST2 performance for IDL 7.1:
|       OS_FAMILY=unix, OS=darwin, ARCH=x86_64
|       Sun Sep 27 20:28:25 2009
       1    0.0207169 Empty For loop,     2000000 times
       2    0.0126920 Call empty procedure (1 param) 100,000 times
       3   0.00679803 Add 100,000 integer scalars and store
       4   0.00896096 25,000 scalar loops each of 5 ops, 2 =, 1 if)
       5   0.00172901 Mult 512 by 512 byte by constant and store, 10
times
       6   0.00581622 Shift 512 by 512 byte and store, 100 times
       7   0.00864697 Add constant to 512 x 512 byte array and store,
50 times
       8   0.00513697 Add two 512 by 512 byte images and store, 30
times
       9   0.00754595 Mult 512 by 512 floating by constant and store,
30 times
      10   0.00503111 Add constant to 512 x 512 floating and store, 40
times
      11   0.00861907 Add two 512 by 512 floating images and store, 30
times
      12   0.00292897 Generate 225000 random numbers
      13   0.00538015 Invert a 150 by 150 random matrix
      14   0.00166893 LU Decomposition of a 150 by 150 random matrix
      15   0.00734806 Transpose 256 x 256 byte, FOR loops
      16    0.0101581 Transpose 256 x 256 byte, row and column ops x
10
      17   0.00205112 Transpose 256 x 256 byte, TRANSPOSE function x
10
      18    0.0148380 Log of 100,000 numbers, FOR loop
      19  0.000674009 Log of 100,000 numbers, vector ops
```

20 0.0147550 131072 point forward plus inverse FFT
21 0.0357552 Smooth 512 by 512 byte array, 5x5 boxcar, 10 times
22 0.00423193 Smooth 512 by 512 floating array, 5x5 boxcar, 2 times
23 0.00666094 Write and read 512 by 512 byte array x 20
0.198143=Total Time, 0.0061516169=Geometric mean, 23 tests.
