Subject: Re: time test MacOC
Posted by gurman on Sun, 15 Mar 1998 08:00:00 GMT
View Forum Message <> Reply to Message

In article <pford-1403981814480001@news.texas.net>, pford@bcm.tmc.edu (Patrick Ford, MD) wrote:

> These are the time test results for a PPC upgraded with a PowerLogix
> board, G# processor 308 MHz, 1 Mbyte backside cache 1:1, 44MHz system bus,
> MacOS 8.1 with many extensions. I could probably tweak the speed somewhat
> by using only one display card and a better display card and SCSI 3 card
> and drives. I am using inexpensive SCSI 1 drives.
>
[individual test results snipped]
> 3.43333=Total Time, 0.099091732=Geometric mean, 23 tests.
> IDL> TIME_TEST2
> 14.5333=Total Time, 0.24510466=Geometric mean, 23 tests.
> IDL> TIME_TEST3
> 26.0333=Total Time, 0.47938677=Geometric mean, 23 tests.
>

FWIW, the following are IDL 5.0.2 time test results on a Power Computing Poer Center Pro 210 (210 MHz 604e, 60 MHz bus speed, 1 Mbyte L2 cache, cheap Ultra SCSI narrow drive):

```
IDL> time_test
|TIME_TEST performance for IDL 5.0.2:
|      OS_FAMILY=MacOS, OS=MacOS, ARCH=PowerMac
      1     0.250000 Empty For loop, 1 million times
      2     0.316667 Call empty procedure (1 param) 100,000 times
      3     0.150000 Add 100,000 integer scalars and store
      4     0.183333 25,000 scalar loops each of 5 ops, 2 =, 1 if)
      5     0.116667 Mult 512 by 512 byte by constant and store, 10 times
      6     0.116667 Shift 512 by 512 byte and store, 10 times
      7     0.133333 Add constant to 512 x 512 byte array and store, 10 times
      8     0.216667 Add two 512 by 512 byte images and store, 10 times
      9     0.566667 Mult 512 by 512 floating by constant and store, 10 times
     10      0.633333 Add constant to 512 x 512 floating and store, 10 times
     11      0.816667 Add two 512 by 512 floating images and store, 10 times
     12     0.0666667 Invert a 100 by 100 random matrix
     13      0.200000 Transpose 256 x 256 byte, FOR loops
     14     0.0166667 Transpose 256 x 256 byte, row and column ops
     15     0.0166667 Transpose 256 x 256 byte, transpose function
     16      0.466667 Log of 100,000 numbers, FOR loop
     17     0.0500000 Log of 100,000 numbers, vector ops
     18      0.400000 Add two 100000 element floating vectors, FOR loop
     19     0.0166667 Add two 100000 element floating vectors, vector op
     20      0.150000 65536 point real to complex FFT
```

```
      21    0.0500000 Smooth 512 by 512 byte array, 5x5 boxcar
      22    0.0666667 Smooth 512 by 512 floating array, 5x5 boxcar
      23     0.466667 Write and read 10 512 by 512 byte arrays
      5.46667=Total Time,       0.14236112=Geometric mean,      23 tests.
IDL> time_test2
|TIME_TEST2 performance for IDL 5.0.2:
|       OS_FAMILY=MacOS, OS=MacOS, ARCH=PowerMac
|  Sun Mar 15 01:00:42 1998
       1     0.466667 Empty For loop,     2000000 times
       2     0.316667 Call empty procedure (1 param) 100,000 times
       3     0.166667 Add 100,000 integer scalars and store
       4     0.183333 25,000 scalar loops each of 5 ops, 2 =, 1 if)
       5     0.133333 Mult 512 by 512 byte by constant and store, 10 times
       6      1.23333 Shift 512 by 512 byte and store, 100 times
       7     0.683333 Add constant to 512 x 512 byte array and store, 50 times
       8     0.733333 Add two 512 by 512 byte images and store, 30 times
       9      1.68333 Mult 512 by 512 floating by constant and store, 30 times
      10      1.76667 Add constant to 512 x 512 floating and store, 40 times
      11      2.88333 Add two 512 by 512 floating images and store, 30 times
      12     0.116667 Generate 225000 random numbers
      13     0.250000 Invert a 150 by 150 random matrix
      14    0.0500001 LU Decomposition of a 150 by 150 random matrix
      15     0.266667 Transpose 256 x 256 byte, FOR loops
      16     0.250000 Transpose 256 x 256 byte, row and column ops x 10
      17     0.250000 Transpose 256 x 256 byte, TRANSPOSE function x 10
      18     0.400000 Log of 100,000 numbers, FOR loop
      19    0.0500001 Log of 100,000 numbers, vector ops
      20      1.01667       131072 point forward plus inverse FFT
      21     0.450000 Smooth 512 by 512 byte array, 5x5 boxcar, 10 times
      22     0.116667 Smooth 512 by 512 floating array, 5x5 boxcar, 2 times
      23      6.23333 Write and read 512 by 512 byte array x 20
      19.7000=Total Time,       0.39464166=Geometric mean,      23 tests.
IDL> time_test3
|TIME_TEST3 performance for IDL 5.0.2:
|       OS_FAMILY=MacOS, OS=MacOS, ARCH=PowerMac
|  Sun Mar 15 01:01:08 1998
       1     0.483333 Empty For loop, 2000000 times
       2     0.300000 Call empty procedure (1 param) 100000 times
       3     0.333333 Add 200000 integer scalars and store
       4     0.450000 50000 scalar loops each of 5 ops, 2 =, 1 if)
       5     0.383333 Mult 512 by 512 byte by constant and store, 30 times
       6      3.55000 Shift 512 by 512 byte and store, 300 times
       7      1.30000 Add constant to 512x512 byte array, 100 times
       8      1.80000 Add two 512 by 512 byte arrays and store, 80 times
       9      1.71667 Mult 512 by 512 floating by constant, 30 times
      10      3.51667 Shift 512 x 512 array, 60 times
      11      2.88333 Add two 512 by 512 floating images, 40 times
      12     0.516667 Generate 1000000 random numbers
```

```
 13     0.483333 Invert a 192^2 random matrix
 14     0.166667 LU Decomposition of a 192^2 random matrix
 15     0.466667 Transpose 384^2 byte, FOR loops
 16     0.650000 Transpose 384^2 byte, row and column ops x 10
 17      5.85000 Transpose 384^2 byte, TRANSPOSE function x 100
 18     0.433333 Log of 100000 numbers, FOR loop
 19     0.550000 Log of 100000 numbers, vector ops 10 times
 20      1.01667 131072 point forward plus inverse FFT
 21     0.516667 Smooth 512 by 512 byte array, 5x5 boxcar, 10 times
 22     0.350000 Smooth 512 by 512 floating array, 5x5 boxcar, 5 times
 23      14.0500 Write and read 512 by 512 byte array x 40
41.7667=Total Time,       0.88139225=Geometric mean,      23 tests.
```

The improvement with the G3 appears to be almost in proportion to the ratio of the clock speeds, but there are too many variables here (e.g. backside vs. motherboard L2, different IDL versions) to draw simple conclusions.... other than that I'll keep that $1800 in my pocket for now, thanks.

Now if they come out with a 400 MHz copper G4 card next year that will work in the PC Pro..... I know, dream on.

Cheers,

Joe Gurman

--
Joseph B. Gurman / NASA Goddard Space Flight Center/ Solar Data Analysis Center / Code 682 / Greenbelt MD 20771 USA / Federal employees are still prohibited from holding opinions while at work. Any opinions expressed herein must therefore be someone else's.

Subject: Re: time test MacOC
Posted by uk2 on Mon, 16 Mar 1998 08:00:00 GMT

View Forum Message <> Reply to Message

Hi Guys,

Here are results from a Dell workstation 333Mz/384Mb Ram running win NT.
While I love macs and UNIX machines, these numbers are pretty impressive.
Can anyone supply faster numbers on another platform?

-Pete Riley

```
IDL> time_test
% Compiled module: TIME_TEST.
```

```
TIME_TEST is obsolete.
Use the newer, more accurate, TIME_TEST2, contained in this file.
|TIME_TEST performance for IDL 5.1 Beta 2:
|       OS_FAMILY=Windows, OS=Win32, ARCH=x86
       1     0.100000 Empty For loop, 1 million times
       2     0.170000 Call empty procedure (1 param) 100,000 times
       3     0.110000 Add 100,000 integer scalars and store
       4    0.0999999 25,000 scalar loops each of 5 ops, 2 =, 1 if)
       5    0.0400000 Mult 512 by 512 byte by constant and store, 10 times
       6    0.0300000 Shift 512 by 512 byte and store, 10 times
       7    0.0510000 Add constant to 512 x 512 byte array and store, 10 times
       8    0.0800000 Add two 512 by 512 byte images and store, 10 times
       9     0.180000 Mult 512 by 512 floating by constant and store, 10 times
      10     0.160000 Add constant to 512 x 512 floating and store, 10 times
      11     0.221000 Add two 512 by 512 floating images and store, 10 times
      12    0.0400000 Invert a 100 by 100 random matrix
      13     0.110000 Transpose 256 x 256 byte, FOR loops
      14    0.0200000 Transpose 256 x 256 byte, row and column ops
      15   0.00999999 Transpose 256 x 256 byte, transpose function
      16     0.230000 Log of 100,000 numbers, FOR loop
      17    0.0500000 Log of 100,000 numbers, vector ops
      18     0.231000 Add two 100000 element floating vectors, FOR loop
      19    0.0100001 Add two 100000 element floating vectors, vector op
      20    0.0500000 65536 point real to complex FFT
      21    0.0400000 Smooth 512 by 512 byte array, 5x5 boxcar
      22    0.0200000 Smooth 512 by 512 floating array, 5x5 boxcar
      23    0.0500000 Write and read 10 512 by 512 byte arrays
     2.10300=Total Time,      0.063319051=Geometric mean,      23 tests.
IDL> time_test2
|TIME_TEST2 performance for IDL 5.1 Beta 2:
|       OS_FAMILY=Windows, OS=Win32, ARCH=x86
| Mon Mar 16 16:16:15 1998
       1     0.190000 Empty For loop,     2000000 times
       2     0.171000 Call empty procedure (1 param) 100,000 times
       3     0.110000 Add 100,000 integer scalars and store
       4    0.0999999 25,000 scalar loops each of 5 ops, 2 =, 1 if)
       5    0.0500001 Mult 512 by 512 byte by constant and store, 10 times
       6     0.371000 Shift 512 by 512 byte and store, 100 times
       7     0.250000 Add constant to 512 x 512 byte array and store, 50 times
       8     0.260000 Add two 512 by 512 byte images and store, 30 times
       9     0.551000 Mult 512 by 512 floating by constant and store, 30 times
      10     0.491000 Add constant to 512 x 512 floating and store, 40 times
      11     0.851000 Add two 512 by 512 floating images and store, 30 times
      12    0.0400001 Generate 225000 random numbers
      13     0.120000 Invert a 150 by 150 random matrix
      14    0.0300000 LU Decomposition of a 150 by 150 random matrix
      15     0.101000 Transpose 256 x 256 byte, FOR loops
      16     0.190000 Transpose 256 x 256 byte, row and column ops x 10
```

```
      17    0.0800000 Transpose 256 x 256 byte, TRANSPOSE function x 10
      18     0.220000 Log of 100,000 numbers, FOR loop
      19    0.0400000 Log of 100,000 numbers, vector ops
      20     0.551000       131072 point forward plus inverse FFT
      21     0.391000 Smooth 512 by 512 byte array, 5x5 boxcar, 10 times
      22    0.0500000 Smooth 512 by 512 floating array, 5x5 boxcar, 2 times
      23     0.120000 Write and read 512 by 512 byte array x 20
      5.32800=Total Time,       0.15530169=Geometric mean,      23 tests.
IDL> time_test3
|TIME_TEST3 performance for IDL 5.1 Beta 2:
|       OS_FAMILY=Windows, OS=Win32, ARCH=x86
| Mon Mar 16 16:16:24 1998
       1     0.190000 Empty For loop, 2000000 times
       2     0.171000 Call empty procedure (1 param) 100000 times
       3     0.230000 Add 200000 integer scalars and store
       4     0.200000 50000 scalar loops each of 5 ops, 2 =, 1 if)
       5     0.151000 Mult 512 by 512 byte by constant and store, 30 times
       6      1.13100 Shift 512 by 512 byte and store, 300 times
       7     0.501000 Add constant to 512x512 byte array, 100 times
       8     0.681000 Add two 512 by 512 byte arrays and store, 80 times
       9     0.551000 Mult 512 by 512 floating by constant, 30 times
      10     0.821000 Shift 512 x 512 array, 60 times
      11     0.961000 Add two 512 by 512 floating images, 40 times
      12     0.171000 Generate 1000000 random numbers
      13     0.250000 Invert a 192^2 random matrix
      14    0.0900000 LU Decomposition of a 192^2 random matrix
      15     0.260000 Transpose 384^2 byte, FOR loops
      16     0.501000 Transpose 384^2 byte, row and column ops x 10
      17      1.73300 Transpose 384^2 byte, TRANSPOSE function x 100
      18     0.220000 Log of 100000 numbers, FOR loop
      19     0.441000 Log of 100000 numbers, vector ops 10 times
      20     0.540000 131072 point forward plus inverse FFT
      21     0.401000 Smooth 512 by 512 byte array, 5x5 boxcar, 10 times
      22     0.140000 Smooth 512 by 512 floating array, 5x5 boxcar, 5 times
      23     0.841000 Write and read 512 by 512 byte array x 40
      11.1760=Total Time,       0.36417970=Geometric mean,      23 tests.
```

Subject: Re: time test MacOC
Posted by harmonic on Mon, 23 Mar 1998 08:00:00 GMT
View Forum Message <> Reply to Message

In article <6ekfpvarl1@nnrp1.dejanews.com>, uk2@scibernet.com says...
Could you post/send the source for time_test? (If it is freely
distributable). I am curious to compare these benchmarks to O-Matrix.

> Hi Guys,
>
> Here are results from a Dell workstation 333Mz/384Mb Ram running win NT.
> While I love macs and UNIX machines, these numbers are pretty impressive.
> Can anyone supply faster numbers on another platform?
>
> -Pete Riley
>
>
> IDL> time_test
> % Compiled module: TIME_TEST.
> TIME_TEST is obsolete.
> Use the newer, more accurate, TIME_TEST2, contained in this file.
> |TIME_TEST performance for IDL 5.1 Beta 2:
> | OS_FAMILY=Windows, OS=Win32, ARCH=x86
> 1 0.100000 Empty For loop, 1 million times
> 2 0.170000 Call empty procedure (1 param) 100,000 times
> 3 0.110000 Add 100,000 integer scalars and store
> 4 0.0999999 25,000 scalar loops each of 5 ops, 2 =, 1 if)
> 5 0.0400000 Mult 512 by 512 byte by constant and store, 10 times
> 6 0.0300000 Shift 512 by 512 byte and store, 10 times
> 7 0.0510000 Add constant to 512 x 512 byte array and store, 10 times
> 8 0.0800000 Add two 512 by 512 byte images and store, 10 times
> 9 0.180000 Mult 512 by 512 floating by constant and store, 10 times
> 10 0.160000 Add constant to 512 x 512 floating and store, 10 times
> 11 0.221000 Add two 512 by 512 floating images and store, 10 times
> 12 0.0400000 Invert a 100 by 100 random matrix
> 13 0.110000 Transpose 256 x 256 byte, FOR loops
> 14 0.0200000 Transpose 256 x 256 byte, row and column ops
> 15 0.00999999 Transpose 256 x 256 byte, transpose function
> 16 0.230000 Log of 100,000 numbers, FOR loop
> 17 0.0500000 Log of 100,000 numbers, vector ops
> 18 0.231000 Add two 100000 element floating vectors, FOR loop
> 19 0.0100001 Add two 100000 element floating vectors, vector op
> 20 0.0500000 65536 point real to complex FFT
> 21 0.0400000 Smooth 512 by 512 byte array, 5x5 boxcar
> 22 0.0200000 Smooth 512 by 512 floating array, 5x5 boxcar
> 23 0.0500000 Write and read 10 512 by 512 byte arrays
> 2.10300=Total Time, 0.063319051=Geometric mean, 23 tests.
> IDL> time_test2
> |TIME_TEST2 performance for IDL 5.1 Beta 2:
> | OS_FAMILY=Windows, OS=Win32, ARCH=x86
> | Mon Mar 16 16:16:15 1998
> 1 0.190000 Empty For loop, 2000000 times
> 2 0.171000 Call empty procedure (1 param) 100,000 times
> 3 0.110000 Add 100,000 integer scalars and store
> 4 0.0999999 25,000 scalar loops each of 5 ops, 2 =, 1 if)

> 5 0.0500001 Mult 512 by 512 byte by constant and store, 10 times
> 6 0.371000 Shift 512 by 512 byte and store, 100 times
> 7 0.250000 Add constant to 512 x 512 byte array and store, 50 times

--
Beau Paisley / Harmonic Software
Makers of O-Matrix - A High-Performance Data Analysis Environment
http://www.omatrix.com - for free copy of O-Matrix Light
