Subject: Re: AMD CPU - (pentium way faster)
Posted by Mark Hadfield on Fri, 01 Feb 2002 00:36:12 GMT
View Forum Message <> Reply to Message

"Robert Stockwell" <rgs1967@hotmail.com> wrote in message
news:3C59CF8F.60804@hotmail.com...
> Mark Hadfield wrote:
>> I have been comparing Linux & Windows 2000 on the same machine:
>> Pentium 3 800 MHz, 384 MB RAM. Linux wins the time_test2 comparison by
>> about 10 % (though Windows tries to cheat on the geometric mean by
>> recording a zero time for one of the tests).
>
> Hmmm the exact opposite of my machine (win2000 is faster than linux).
> What flavour of Linux was it? Also, what IDL version?

Before I answer those questions, below my sig are output files from time_test3. This is, I think, the best of the time_test family to use on modern machines. You will see that Linux and Windows are very close; Windows wins by a narrow margin on geometric mean whereas Linux wins by a narrow margin on geometric mean. There's some interesting variation in relative performance between the test. Procedure was to run time_test3 several times on a lightly loaded machine until results stabilised.

Configuration: Compaq DeskPro EN, Pentium 3 800 MHz, 384 MiB RAM, IDL 5.5 (using IDLDE on both machines), Windows 2000 vs RedHat Linux 7.2 with kernel 2.4.7-10.

Mark Hadfield
m.hadfield@niwa.co.nz http://katipo.niwa.co.nz/~hadfield
National Institute for Water and Atmospheric Research

```
; IDL Version 5.5 (linux x86)
; Journal File for hadfield@hadfield
; Working directory: /home/hadfield/idl/mghlib/_Motley
; Date: Fri Feb  1 13:08:31 2002
time_test3
;|TIME_TEST3 performance for IDL 5.5:
;|      OS_FAMILY=unix, OS=linux, ARCH=x86
;| Fri Feb  1 13:08:34 2002
;       1     0.105139 Empty For loop, 2000000 times
;       2     0.103525 Call empty procedure (1 param) 100000 times
;       3     0.127712 Add 200000 integer scalars and store
;       4     0.119736 50000 scalar loops each of 5 ops, 2 =, 1 if)
;       5     0.148752 Mult 512 by 512 byte by constant and store, 30 times
;       6     0.896322 Shift 512 by 512 byte and store, 300 times
;       7     0.501356 Add constant to 512x512 byte array, 100 times
;       8     0.494077 Add two 512 by 512 byte arrays and store, 80 times
```

```
;       9     0.404303 Mult 512 by 512 floating by constant, 30 times
;      10     0.677836 Shift 512 x 512 array, 60 times
;      11     0.667641 Add two 512 by 512 floating images, 40 times
;      12     0.164803 Generate 1000000 random numbers
;      13    0.0913930 Invert a 192^2 random matrix
;      14    0.0264250 LU Decomposition of a 192^2 random matrix
;      15     0.128462 Transpose 384^2 byte, FOR loops
;      16     0.175594 Transpose 384^2 byte, row and column ops x 10
;      17     0.307561 Transpose 384^2 byte, TRANSPOSE function x 100
;      18     0.125423 Log of 100000 numbers, FOR loop
;      19     0.155121 Log of 100000 numbers, vector ops 10 times
;      20     0.433018 131072 point forward plus inverse FFT
;      21     0.188205 Smooth 512 by 512 byte array, 5x5 boxcar, 10 times
;      22     0.111943 Smooth 512 by 512 floating array, 5x5 boxcar, 5 times
;      23     0.295521 Write and read 512 by 512 byte array x 40
;      6.44987=Total Time,       0.20485736=Geometric mean,      23 tests.

; IDL Version 5.5, Microsoft Windows (Win32 x86)
; Journal File for hadfield@HADFIELD
; Working directory: C:\Documents and Settings\hadfield
; Date: Fri Feb 01 12:43:36 2002
time_test3
;|TIME_TEST3 performance for IDL 5.5:
;|       OS_FAMILY=Windows, OS=Win32, ARCH=x86
;| Fri Feb 01 12:44:59 2002
;       1    0.0780001 Empty For loop, 2000000 times
;       2    0.0630000 Call empty procedure (1 param) 100000 times
;       3    0.0940000 Add 200000 integer scalars and store
;       4    0.0780001 50000 scalar loops each of 5 ops, 2 =, 1 if)
;       5     0.109000 Mult 512 by 512 byte by constant and store, 30 times
;       6      1.01500 Shift 512 by 512 byte and store, 300 times
;       7     0.359000 Add constant to 512x512 byte array, 100 times
;       8     0.438000 Add two 512 by 512 byte arrays and store, 80 times
;       9     0.469000 Mult 512 by 512 floating by constant, 30 times
;      10     0.844000 Shift 512 x 512 array, 60 times
;      11     0.891000 Add two 512 by 512 floating images, 40 times
;      12    0.0940000 Generate 1000000 random numbers
;      13    0.0780001 Invert a 192^2 random matrix
;      14    0.0159999 LU Decomposition of a 192^2 random matrix
;      15     0.109000 Transpose 384^2 byte, FOR loops
;      16     0.125000 Transpose 384^2 byte, row and column ops x 10
;      17     0.266000 Transpose 384^2 byte, TRANSPOSE function x 100
;      18    0.0930001 Log of 100000 numbers, FOR loop
;      19     0.157000 Log of 100000 numbers, vector ops 10 times
;      20     0.468000 131072 point forward plus inverse FFT
;      21     0.172000 Smooth 512 by 512 byte array, 5x5 boxcar, 10 times
;      22    0.0940000 Smooth 512 by 512 floating array, 5x5 boxcar, 5 times
;      23     0.625000 Write and read 512 by 512 byte array x 40
```

; 6.73500=Total Time, 0.17893937=Geometric mean, 23 tests.