

Vectorización en ARMv8 30237

Multiprocesadores - Grado Ingeniería Informática

Esp. en Ingeniería de Computadores Universidad de Zaragoza

Sergio García Esteban

2-julio-2020

Vectorización en ARMv8

1. ¿Cuántas instrucciones se ejecutan en el bucle interno?

```
for (int i = 0; i < LEN; i++)  
    x[i] = alpha*x[i] + beta
```

Calcula la reducción en el número de instrucciones respecto la versión esc.

versión	icount	reducción(%)	reducción(factor)
esc	6144	0	1.0
vec	1536	75	4.0

Indica muy brevemente cómo has calculado los anteriores valores.

El cuerpo del bucle consta de 6 instrucciones ARM, tanto en escalar como en vectorial, el vector `x[]` tiene 1024 elementos y cada elemento ocupa 32 bits.

La version escalar calcula 1 elemento por iteración, ejecutará 6144 instrucciones.

La version vectorial hace uso de la unidad NEON de cálculo vectorial de 128 bits, calcula el resultado de 4 elementos del vector en cada iteración, por lo que ejecutará 4 veces menos iteraciones.

Nota: todas las versiones han sido compiladas con optimización O3,

las versiones escalares con la opción "-fno-tree-vectorize" y las versiones FMA con la opción "-funsafe-math-optimizations".

```

esc
400868: bd400000    ldr s0, [x0]
40086c: 1e280800    fmul     s0, s0, s8
400870: 1e292800    fadd     s0, s0, s9
400874: bc004400    str s0, [x0], #4
400878: eb13001f    cmp x0, x19
40087c: 54ffff61    b.ne     400868 <scale_shift+0x50>

esc+fma
400870: bd400000    ldr s0, [x0]
400874: 1f082400    fmadd    s0, s0, s8, s9
400878: bc004400    str s0, [x0], #4
40087c: eb13001f    cmp x0, x19
400880: 54ffff81    b.ne     400870 <scale_shift+0x50>

vec
400868: 3dc00000    ldr q0, [x0]
40086c: 6e23dc00    fmul     v0.4s, v0.4s, v3.4s
400870: 4e22d400    fadd     v0.4s, v0.4s, v2.4s
400874: 3c810400    str q0, [x0], #16
400878: eb00027f    cmp x19, x0
40087c: 54ffff61    b.ne     400868 <scale_shift+0x50>

vec+fma
400870: 4ea21c41    mov v1.16b, v2.16b
400874: 3dc00000    ldr q0, [x0]
400878: 4e23cc01    fmla     v1.4s, v0.4s, v3.4s
40087c: 3c810401    str q1, [x0], #16
400880: eb00027f    cmp x19, x0
400884: 54ffff61    b.ne     400870 <scale_shift+0x50>

```

2. A partir de los tiempos de ejecución obtenidos [...], calcula las siguientes métricas para todas las versiones ejecutadas:

- Aceleraciones (*speedups*) de las versiones optimizadas.
- Rendimiento (R) en GFLOPS.
- Rendimiento pico (R_{pico}) teórico de un núcleo (*core*), en GFLOPS.
- Velocidad de ejecución de instrucciones (V_I), en Ginstrucciones por segundo (GIPS).

Indica brevemente cómo has realizado los cálculos.

versión	tiempo(ns)	speed-up	R(GFLOPS)	R _{pico} (GFLOPS)	V _I (GIPS)
esc	718.3	1.0	2.78	3	8.55
esc+fma	795.5	0.9	2.51	3	7.72
vec	184.9	3.8	10.81	12	8.30
vec+fma	203.9	3.5	9.80	12	7.53

Nota: UF:1xNEON Base:2.6GHz y Turbo: 3GHz

tiempo -> medido en ejecución.
 speed-up -> tiempo base / nuevo tiempo
 R -> (FLOPs / tiempo(s))*10⁻⁹
 R_{pico} -> UF * ops/UF * CPU_{freq} (1 UF y 3 GHz turbo)
 V_I -> icount / tiempo

¿La velocidad de ejecución de instrucciones es un buen indicador de rendimiento?

No, ya que todas las intrucciones no son igual de productivas, ni igual de costosas.