

# Laboratorio 1

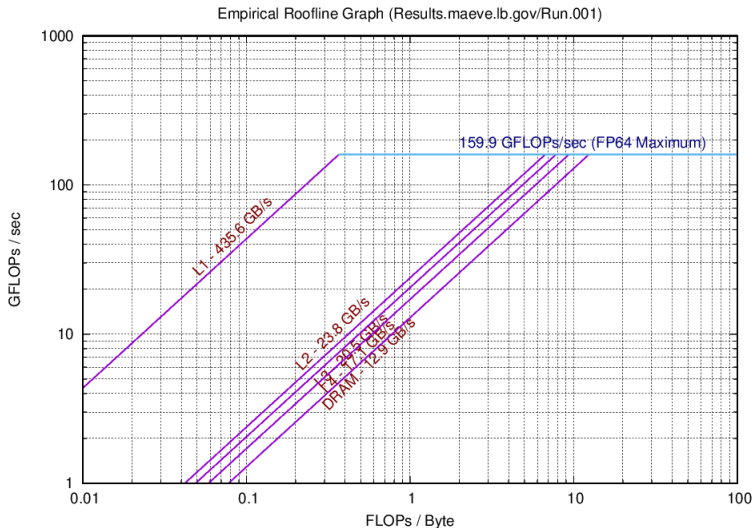
Alvaro Frias Garay - Ary Lautaro Di Bartolo

Universidad Nacional de Córdoba - Universidad de Lautaro

2021

# Características del Hardware y del Software

## Máquina 1: AMD Ryzen 5 2400G with Radeon Vega Graphics



# Características del Hardware y del Software

Compilador: gcc (Ubuntu 9.3.0-17ubuntu1 20.04) 9.3.0

OS: Ubuntu 20.04.2 LTS (Focal Fossa)

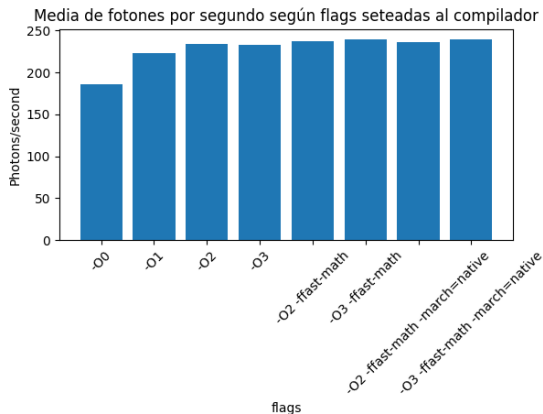
Arquitectura: x86\_64

# Optimizaciones

Primero elegimos un conjunto de flags óptimas

# Optimizaciones

Primero elegimos un conjunto de flags óptimas



# Optimizaciones

Fragmento de código a optimizar

```
do {  
    xi1 = 2.0f * rand() / (float)RAND_MAX - 1.0f;  
    xi2 = 2.0f * rand() / (float)RAND_MAX - 1.0f;  
    t = xi1 * xi1 + xi2 * xi2;  
} while (1.0f < t);
```

# Optimizaciones

¿Qué es lo que hace este código?

# Optimizaciones

¿Qué es lo que hace este código?

$$x_{i1}, x_{i2} \in [-1, 1]$$



# Optimizaciones

¿Qué es lo que hace este código?

$$\begin{aligned} xi1, xi2 &\in [-1, 1] \\ t &= (xi1)^2 + (xi2)^2 \end{aligned}$$

# Optimizaciones

¿Qué es lo que hace este código?

$$\begin{aligned} xi1, xi2 &\in [-1, 1] \\ t &= (xi1)^2 + (xi2)^2 \\ \text{i.e } t &\in [0, 1] \end{aligned}$$

# Optimizaciones

$$t \leq 1$$

# Optimizaciones

$$\begin{aligned} t &\leq 1 \\ (x_{i1})^2 + (x_{i2})^2 &\leq 1 \end{aligned}$$

# Optimizaciones

$$\begin{aligned}t &\leq 1 \\(xi1)^2 + (xi2)^2 &\leq 1 \\(xi2)^2 &\leq 1 - (xi1)^2\end{aligned}$$

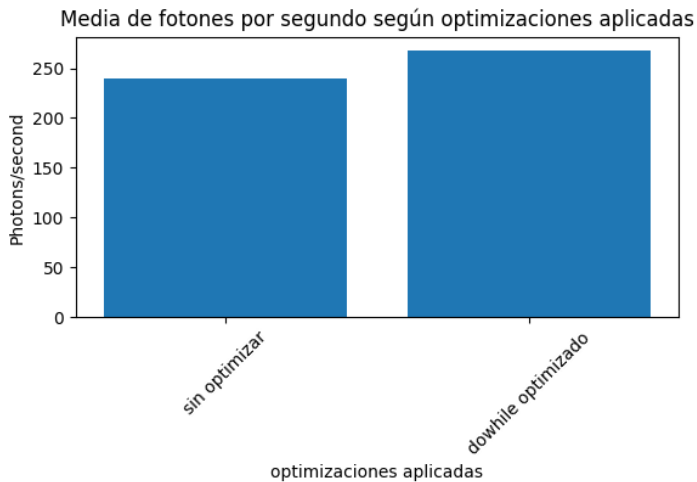
# Optimizaciones

$$\begin{aligned}t &\leq 1 \\(xi1)^2 + (xi2)^2 &\leq 1 \\(xi2)^2 &\leq 1 - (xi1)^2 \\xi2 &\leq \sqrt{1 - (xi1)^2}\end{aligned}$$

# Optimizaciones

$$\begin{aligned}t &\leq 1 \\(xi1)^2 + (xi2)^2 &\leq 1 \\(xi2)^2 &\leq 1 - (xi1)^2 \\xi2 &\leq \sqrt{1 - (xi1)^2} \\&\text{i.e } xi1 \in [0, 1] \\xi2 &\in [0, \sqrt{1 - (xi1)^2}]\end{aligned}$$

# Optimizaciones





# Optimizaciones

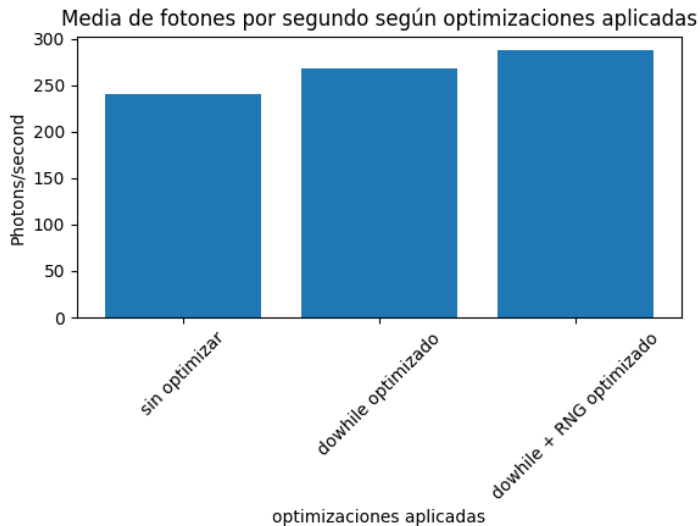
Reemplazamos `rand()` por una implementación de Mersenne Twister en puro C

# Optimizaciones

Reemplazamos `rand()` por una implementación de Mersenne Twister en puro C

<https://github.com/ESultanik/mtwister>

# Optimizaciones



# Optimizaciones

Un par de optimizaciones más;

# Optimizaciones

Un par de optimizaciones más;

- ▶ Reemplazo de división por multiplicación de inverso

# Optimizaciones

Un par de optimizaciones más;

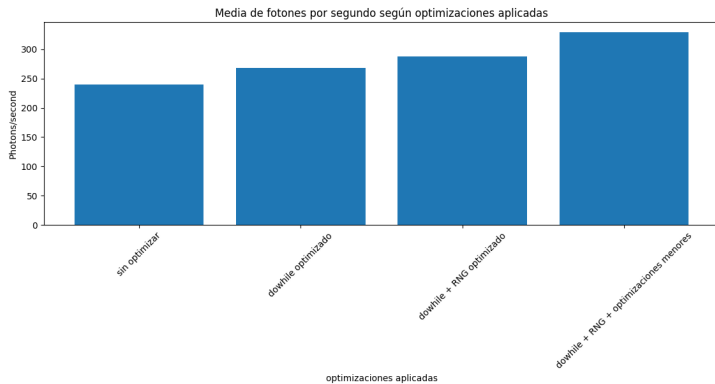
- ▶ Reemplazo de división por multiplicación de inverso  
...Aunque probablemente ya lo hace -ffast-math

# Optimizaciones

Un par de optimizaciones más;

- ▶ Reemplazo de división por multiplicación de inverso  
...Aunque probablemente ya lo hace -ffast-math
- ▶ Añadida flag -flto para inlining de MT

# Optimizaciones





# Scaling

