

# Modelo de Programación GPGPU



José Alejandro Matute



# Que veremos

- Introducción a programación paralela.
- Arquitectura GPU
- Programación CUDA
- Comparación Secuencial vs GPU vs MPI

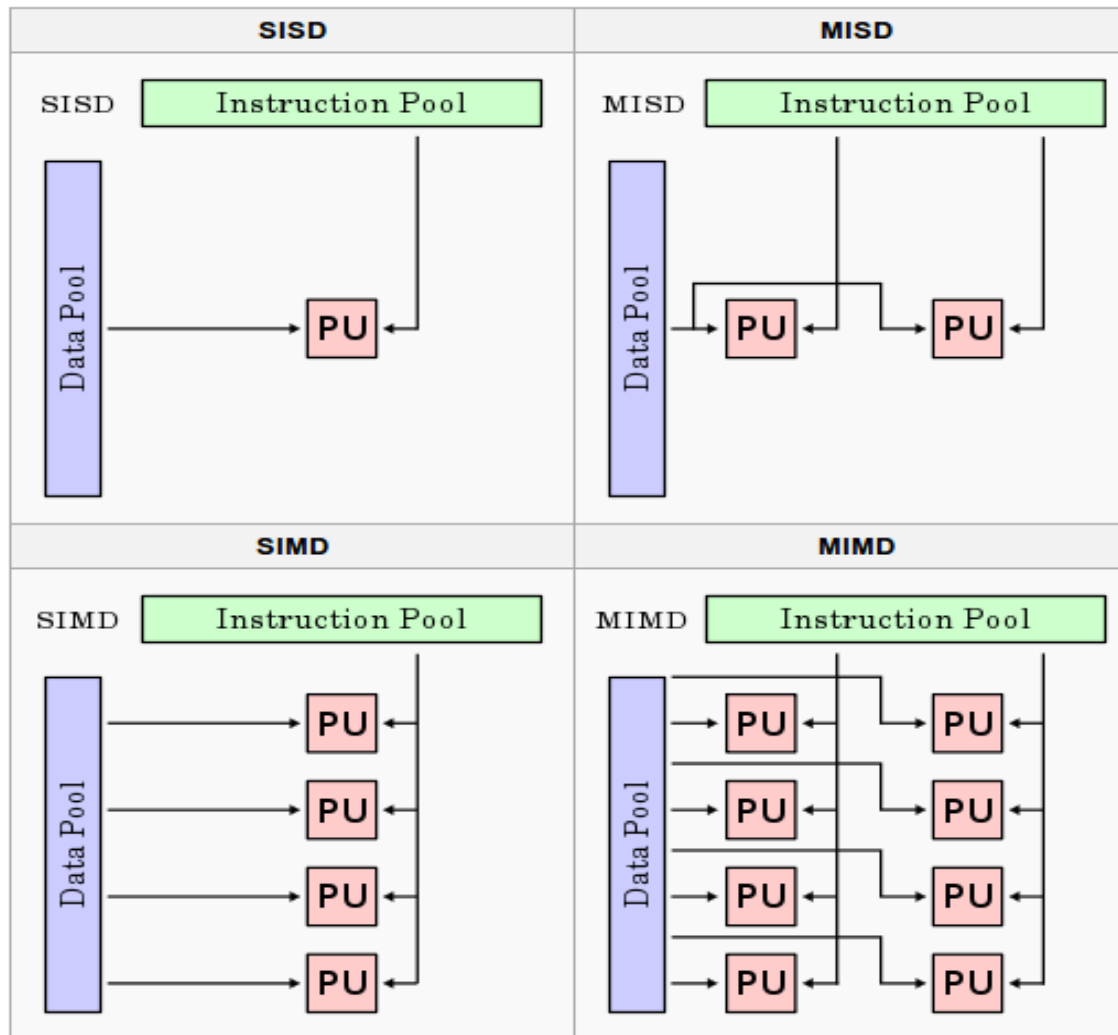


# Que no veremos

- Concurrencia
- Creación de Candados/Banderas/etc..
- Manejo Sección Critica



# Taxonomía de Flynn



# Paralelismo

- Paralelismo de Tareas
  - Partir las tareas para ejecución
- Paralelismo de Datos
  - Partir los datos

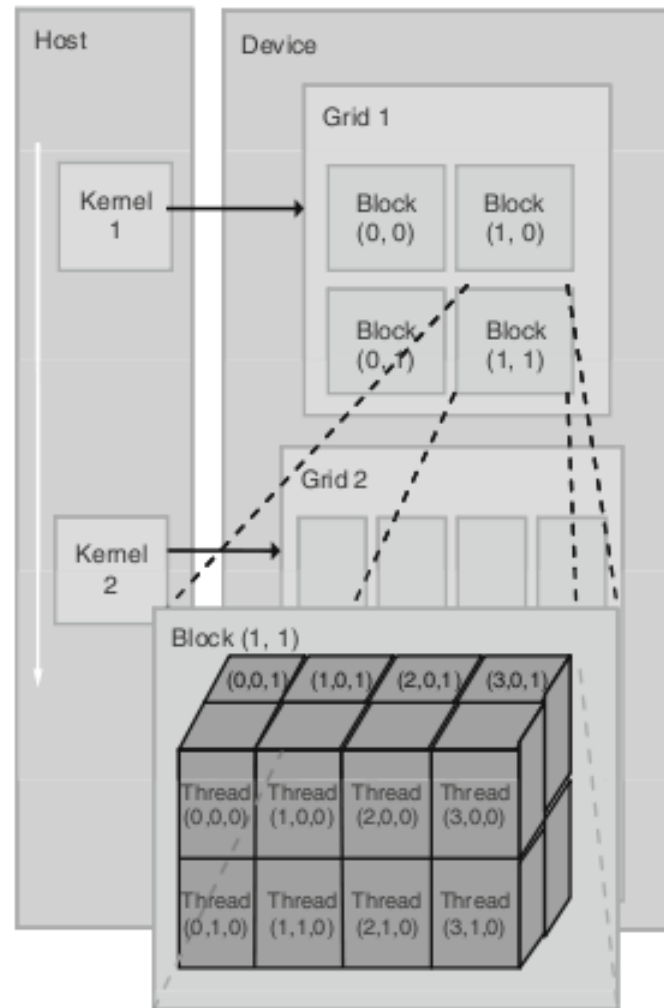


# GPGPU

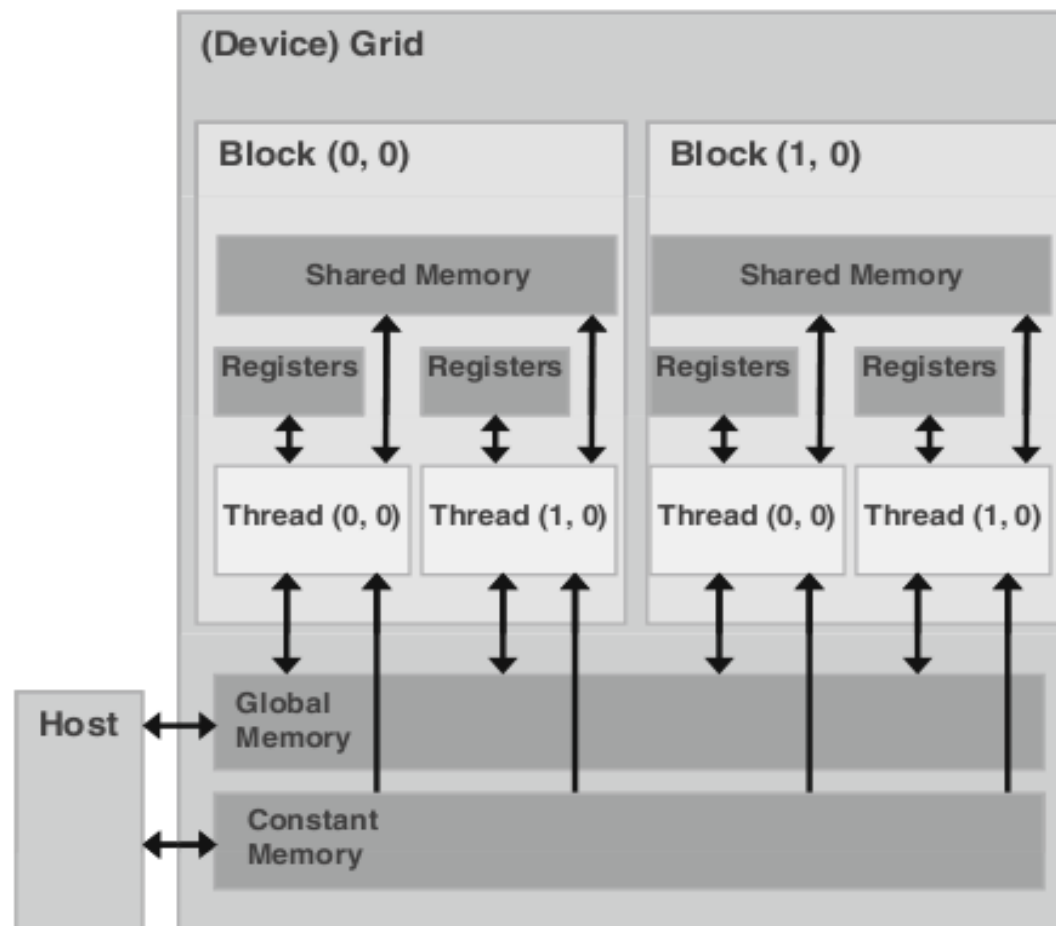
- GPU: Unidad de procesamiento de gráficos
- GPGPU: Utilización de GPU para hacer programación general.
- CUDA: Arquitectura de programación paralela de Nvidia.



# Organización



# Modelo





# Programación CUDA C

```
1
2 __global__ void add(int a,int b,int *c){
3     *c = a+b;
4 }
5
6 int main(){
7     int c;
8     int *resp;
9
10    cudaMalloc( (void**)&resp,sizeof(int));
11
12    add<<<1,1>>>(2,3,resp);
13
14    cudaMemcpy(&c,resp,sizeof(int),cudaMemcpyDeviceToHost);
15
16    cudaFree(resp);
17
18    return c;
19 }
```

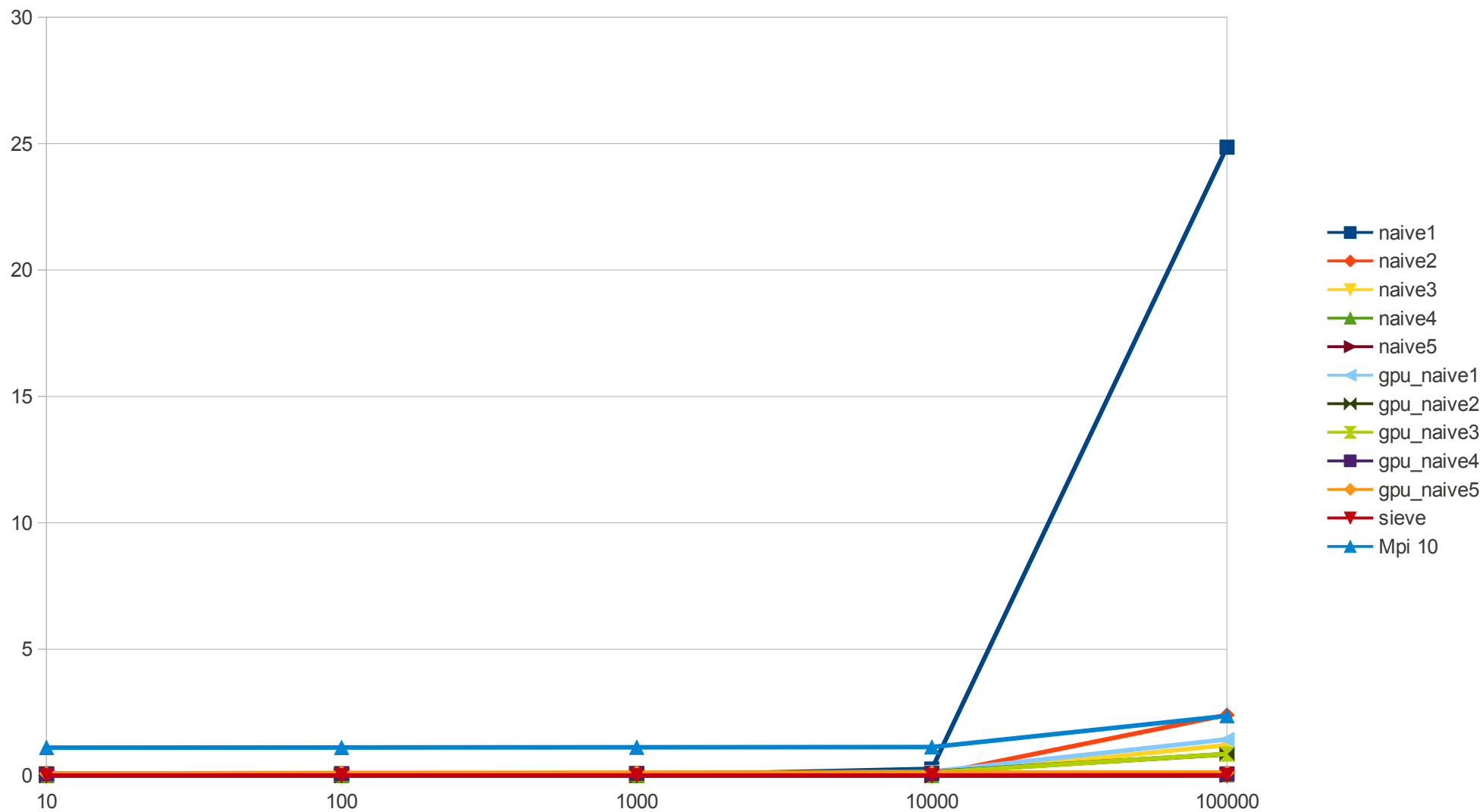


# Problema

- Encontrar la cantidad de números primos de hasta  $N$ .



# Tiempos en segundos

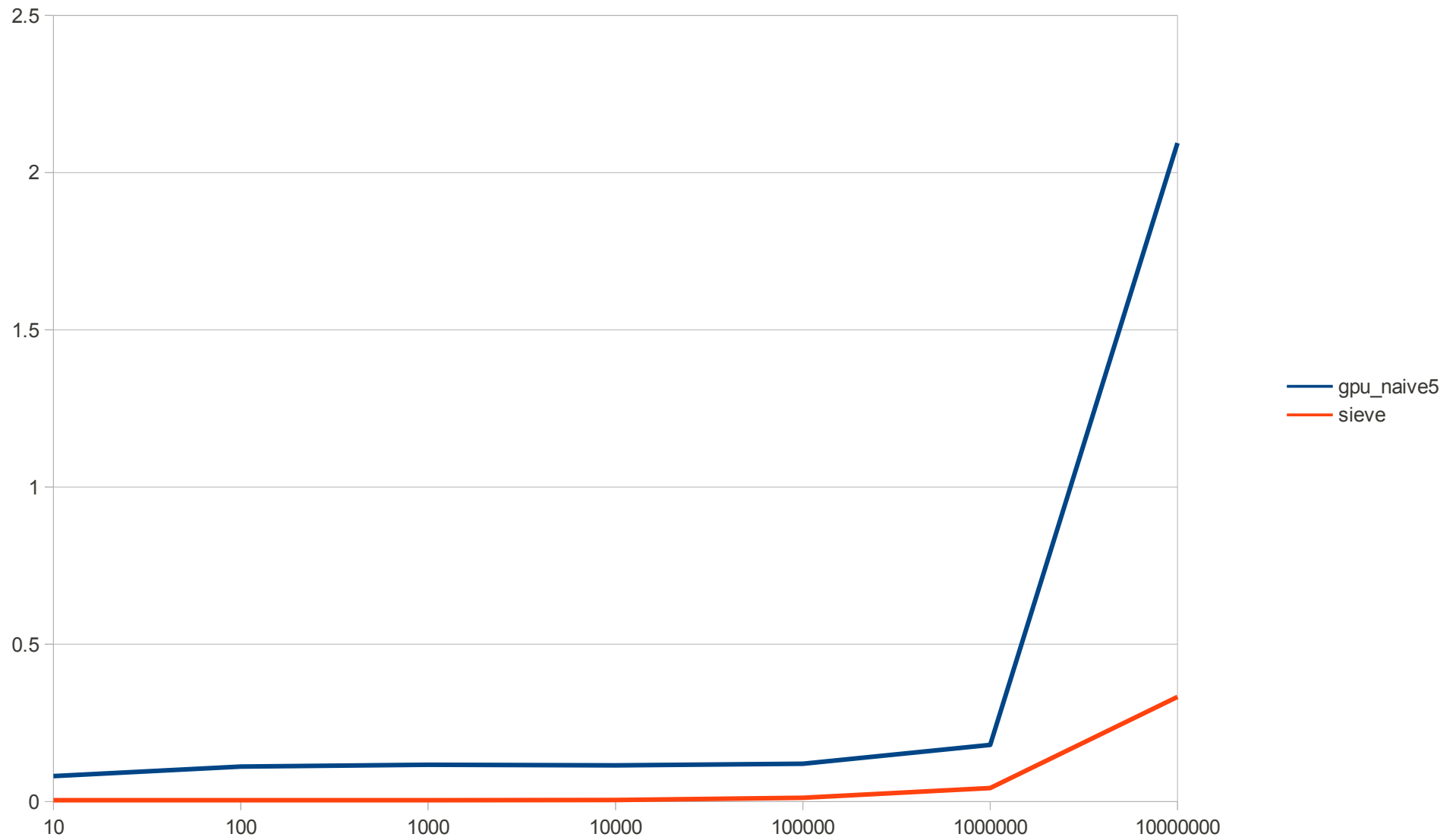


# Magic bullet

- Algunas personas, al encontrarse un problema se dicen a si misma “Ya se, lo haré en paralelo”  
Y ahora tineen sdo problemsa



# Anillo : Dedo :: Herramienta : ?



# Informacion

- MPI: [http://chryswoods.com/beginning\\_mpi](http://chryswoods.com/beginning_mpi)
- Introduction to Parallel Programming *Pacheco*
- The art of multiprocessor programming *Herlihy & Shavit*
- CUDA by example *Sanders & Kandrot*
- Programming massively parallel processors *Kirk & Hwu*
- CUDA PTX [http://www.nvidia.com/content/CUDA-ptx\\_isa\\_1.4.pdf](http://www.nvidia.com/content/CUDA-ptx_isa_1.4.pdf)
- Código y Presentación:  
<https://github.com/jmatuteMI/GPUBarcampHN>

