Clase 01: Introducción

IIC3533 Computación de Alto Rendimiento

Ignacio Labarca – ijlabarca@uc.cl

Departamento de Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Martes 5 de Agosto del 2025

Un poco sobre mí

- Ignacio Labarca – ijlabarca@uc.cl
- Doctorado en Matemáticas Aplicadas, ETH Zurich, Suiza.
- Experiencia práctica en Computación de Alto Rendimiento.
  - Electromagnetismo Computacional

Contenidos del Curso

Sistemas de Cómputo Intensivo

¿Cómo ser más rápido?

¿Cómo paralelizar?

Contenidos del Curso

Sistemas de Cómputo Intensivo

¿Cómo ser más rápido?

¿Cómo paralelizar?

Objetivo General

Punto 1

Utilizar técnicas y conceptos para

- desarrollar
- analizar
- implementar

algoritmos paralelos escalables.

Objetivo General

Punto 2

Entender cómo hacer uso de

tecnologías modernas

de cómputo de alto rendimiento.

Objetivo General

Punto 3

Analizar

problemas actuales y las tecnologías

que permiten soportar este tipo de cómputos.

Contenido

1. Cómputo monoprocesador.
2. Cómputo paralelo.
3. Uso de plataformas de alto rendimiento.
4. Programación con memoria compartida: OpenMP.
5. Programación con paso de mensajes: MPI.
6. Programación en GPUs.
7. Aplicaciones.

Metodología

- Clases de cátedra.
- Sesiones prácticas (trabajo grupal).
- Sesiones de presentación (tres personas).

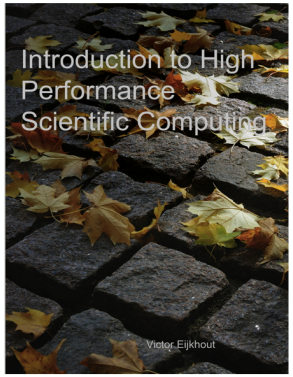
Evaluación

- 4 Controles en clase: 30 % (se elimina el peor).
- 2 Tareas: 30 % (grupales).
- Presentación: 30 %.
- Asistencia a presentaciones: 10 %.

Para aprobar el curso la nota final debe ser mayor o igual a 4,0.

Material Complementario

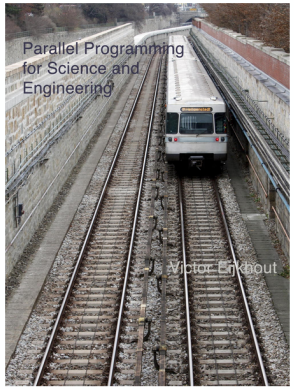
- Libro Guía
  - Victor Eijkhout, "The Science of Computing". University of Texas, 3rd. Edition, 2022.
  - Disponible en: `https://theartofhpc.com`



Volume 1: The Science of Computing

This book has the general background knowledge for scientific computing: computer architecture, parallel computer architecture, computer arithmetic, linear algebra, ODE/PDEs, and how it all comes together in big computations.

Paired with Volume 2, this teaches you all about HPC: the what/why here, and the how over there..



Volume 2: Parallel Programming for Science and Engineering

Much of scientific computing involves parallel programming of some sort. This book is the ultimate introduction to modern versions of MPI and OpenMP, with shorter sections on PETSc, Kokkos, Sycl, Co-array Fortran.

This book is notable for its coverage of MPI and OpenMP in both C, Fortran, C++, and (for MPI) Python.

Volume 3: Introduction to Scientific Programming

Programming in science and engineering is overwhelmingly done in C/C++ and Fortran. This book teaches you modern C++17 (which is really to be preferred over C these days) and Fortran2008.

You can read this as a from-zero introduction to scientific programming, or as a "C++ for C programmers". There are various longer programming projects.

Volume 4: HPC Carpentry

The ecosystem of scientific computing involves much more than just a programming language and a parallel programming system: you need to understand about compilers, build systems, source code control, and a lot more. This book does not aim to be a compendium of all of these, but rather a set of introductions, biased to scientific workflows..

Contenidos del Curso

Sistemas de Cómputo Intensivo

¿Cómo ser más rápido?

¿Cómo paralelizar?

Breve historia del *cómputo*

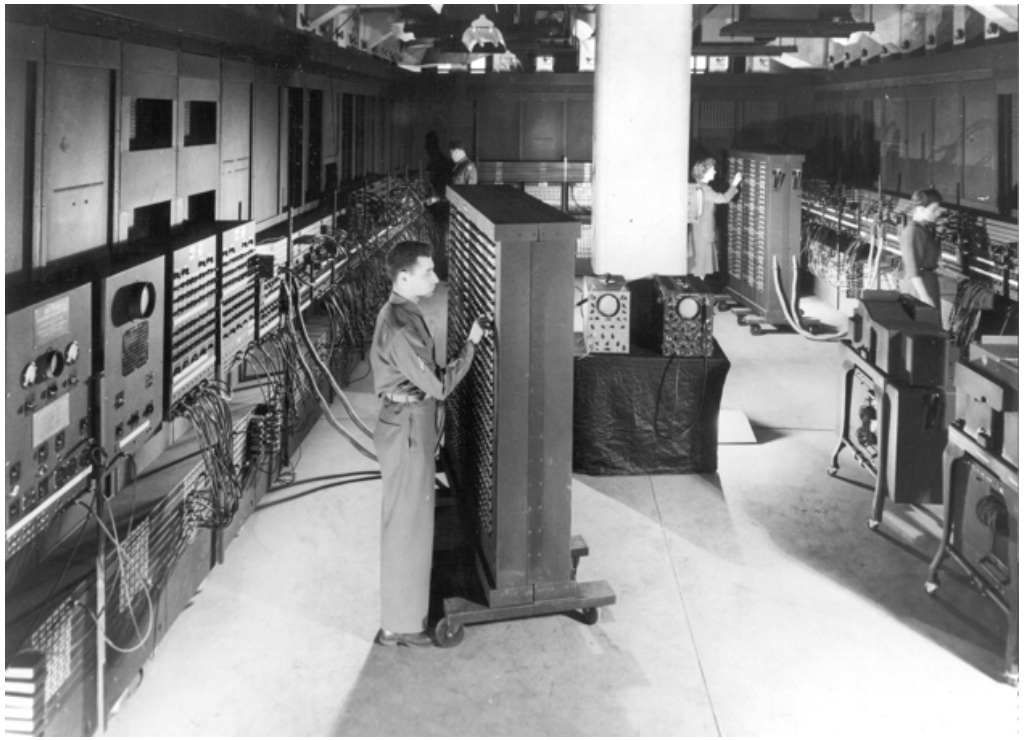
Los primeros *computadores* fueron humanos

El pronóstico del clima

El sueño de Lewis Fry Richardson en 1922:
Weather Prediction by Numerical Process

Computadores electrónicos

Electronic Numerical Integrator and Computer (ENIAC, 1964)



Computadores personales

Apple I kit computer (1976)

Uso cotidiano

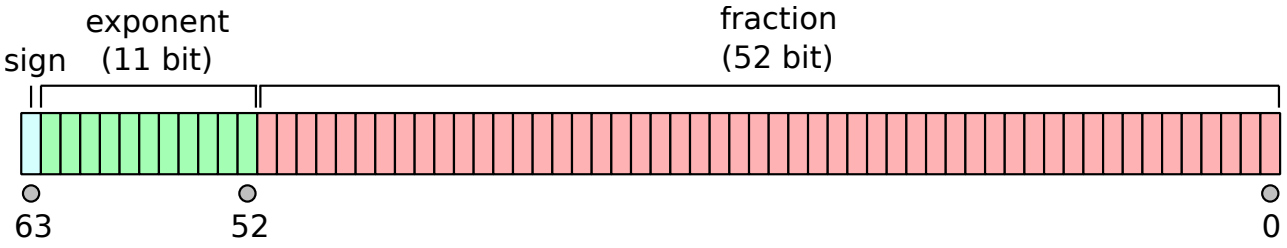
Supercomputadores

El *IBM Roadrunner* logró cómputo a nivel de *petaflops* en 2008

Sistemas de Cómputo Intensivo

Métricas en supercomputadores:

- **FLOP**. Floating Point Operation en precisión doble (8 byte).
- **FLOPS**, o también **FLOP/s**: FLOP por segundo.
- **Byte**. Tamaño de los datos. Precisión doble: 8 byte. Simple: 4 byte.
- **Bit**. Unidad básica. 1 Byte = 8 Bit.



Sistemas de Cómputo Intensivo

https://top500.org/

El computador más rápido del mundo:

El Capitán : cómputos a exaescala

Cores CPU	1,051,392
Cores GPU	9,988,224
Superficie	700 m^2

TOP 500 The List.

HOME LISTS STATISTICS RESOURCES ABOUT MEDIA KIT

El Capitan achieves top spot, Frontier and Aurora follow behind

Nov. 18, 2024

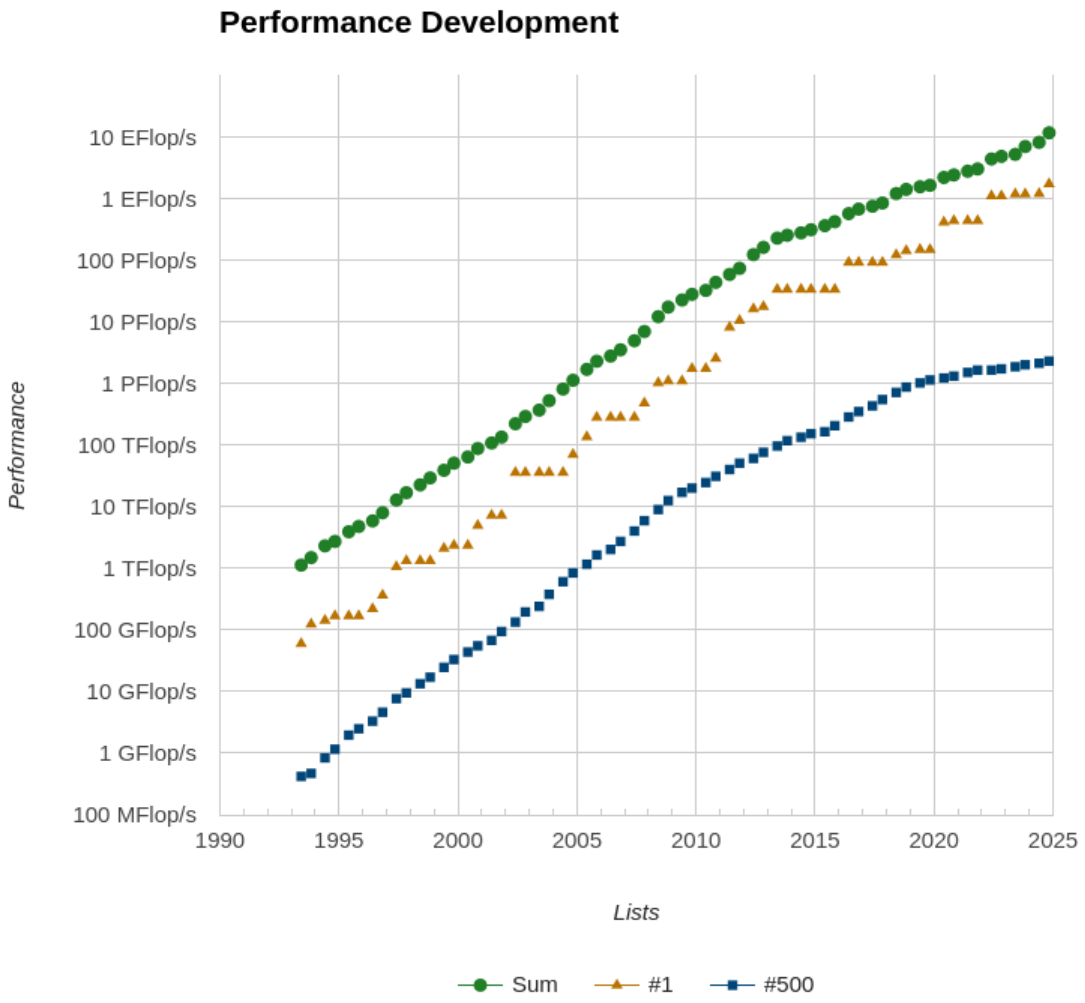
The 64th edition of the TOP500 reveals that El Capitan has achieved the top spot and is officially the third system to reach exascale computing after Frontier and Aurora. Both systems have since moved down to No. 2 and No. 3 spots, respectively. Additionally, new systems have found their way onto the Top 10.

read more »

Sistemas de Cómputo Intensivo

Poder de cómputo se seguirá expandiendo

Mega: 10^6 Giga: 10^9 Tera: 10^{12} Peta: 10^{15} Exa: 10^{18}



Contenidos del Curso

Sistemas de Cómputo Intensivo

¿Cómo ser más rápido?

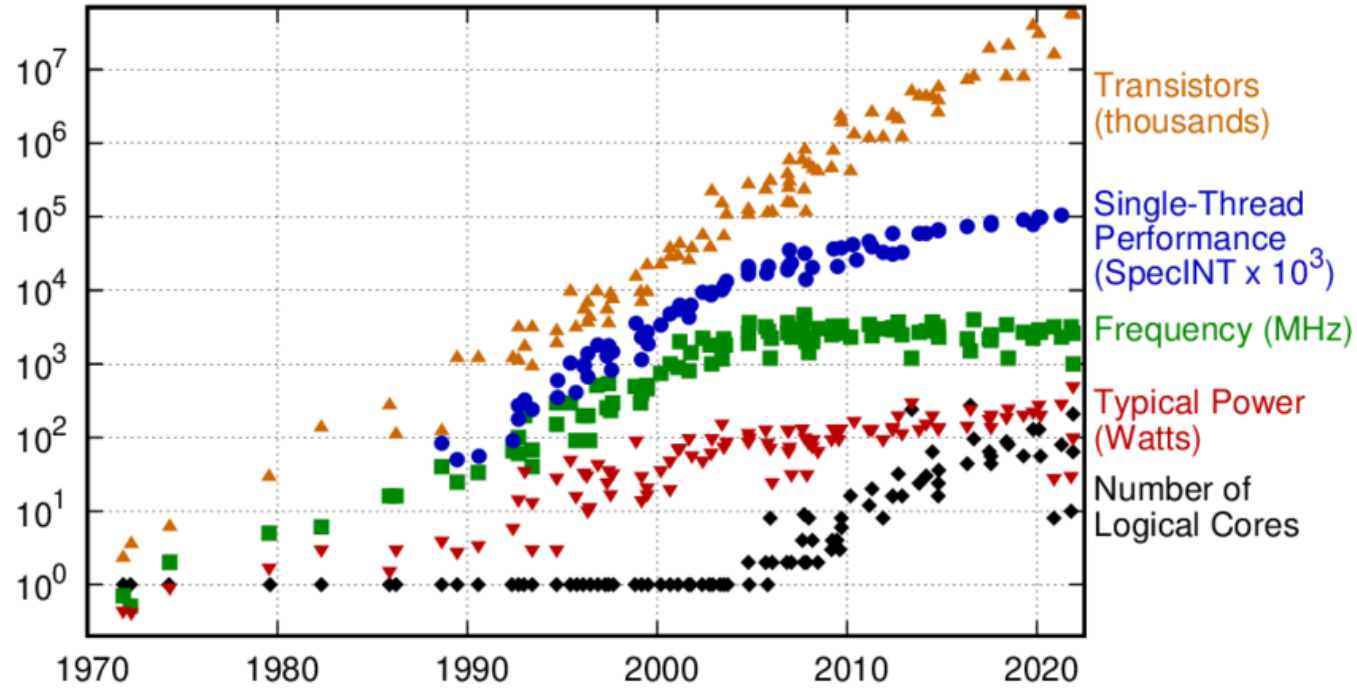
¿Cómo paralelizar?

La Ley de Moore

> *"El número de transistores en un microprocesador se duplicará cada dos años aproximadamente"*.
>
> Gordon Moore, co-fundador Intel, 1975.

Mayor poder para operaciones más complejas .

Ley de Moore



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2021 by K. Rupp

¿Cómo ser más rápido?

Para mayor rapidez, se puede tener

- Más cores en una máquina – *multicore*
  - Dual, Quad, Octa, ...
  - Memoria compartida.
- Más máquinas con múltiples cores
  - *Clusters*.
  - Memoria compartida o distribuida.
- Chips con más procesadores – *manycore* – GPU

¿Paralelismo, distribución, o HPC?

- **Paralelismo**. Cómputo que involucra múltiples tareas ejecutándose simultaneamente como parte de un mismo trabajo.
- **Multiprocesamiento**. Computador ejecutando múltiples programas simultáneamente.
- **HPC**. Cómputo de una tarea muy larga, pero que se ejecuta a la mayor velocidad posible.
- **Cómputo distribuido**. Cómputo que involucra múltiples computadores colaborando en una tarea *complicada*.

¿Para qué paralelizar?

Se presentan desafíos en computación de alto rendimiento

- Tenemos más capacidad de generar datos que de almacenarlos
  - Modelamiento Climático
  - Astronomía
  - Redes sociales
  - Transacciones
- Uso eficiente de unidades de cómputo.
- Uso eficiente de la red que las une.

Contenidos del Curso

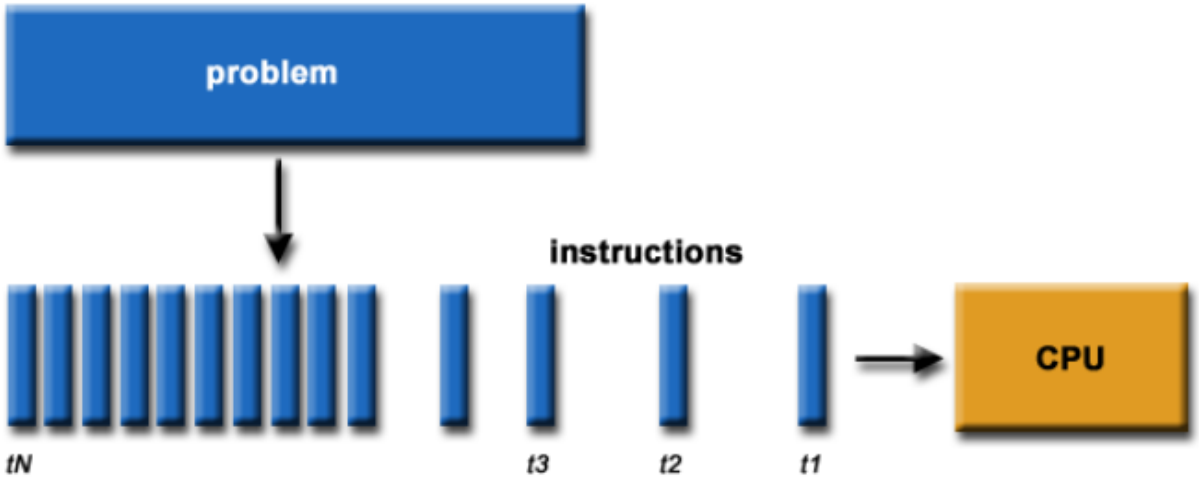
Sistemas de Cómputo Intensivo

¿Cómo ser más rápido?

¿Cómo paralelizar?

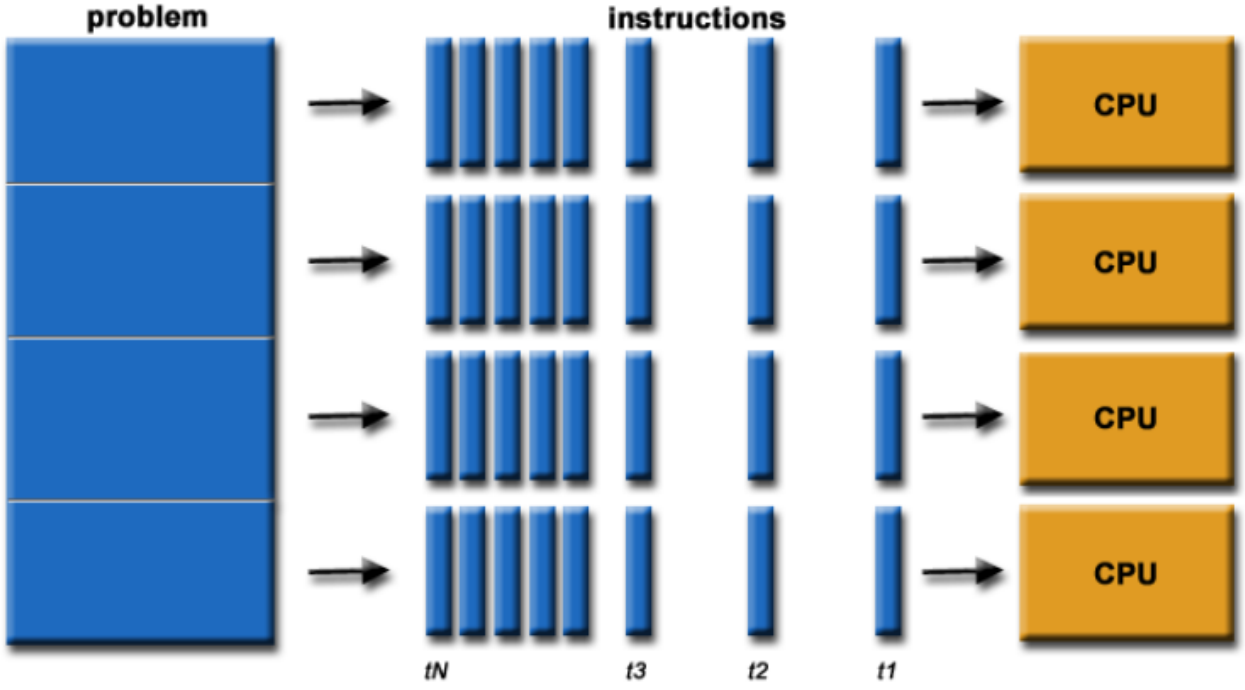
¿Cómo paralelizar?

Problema serial



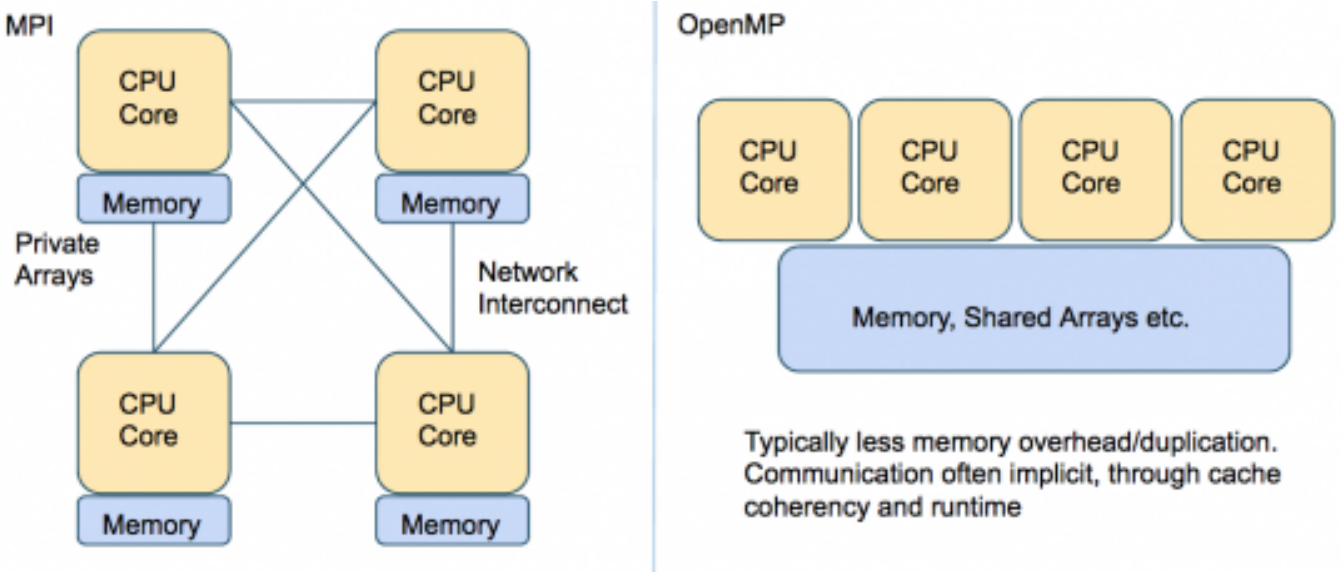
¿Cómo paralelizar?

Problema paralelo



¿Cómo paralelizar?

MPI vs OpenMP



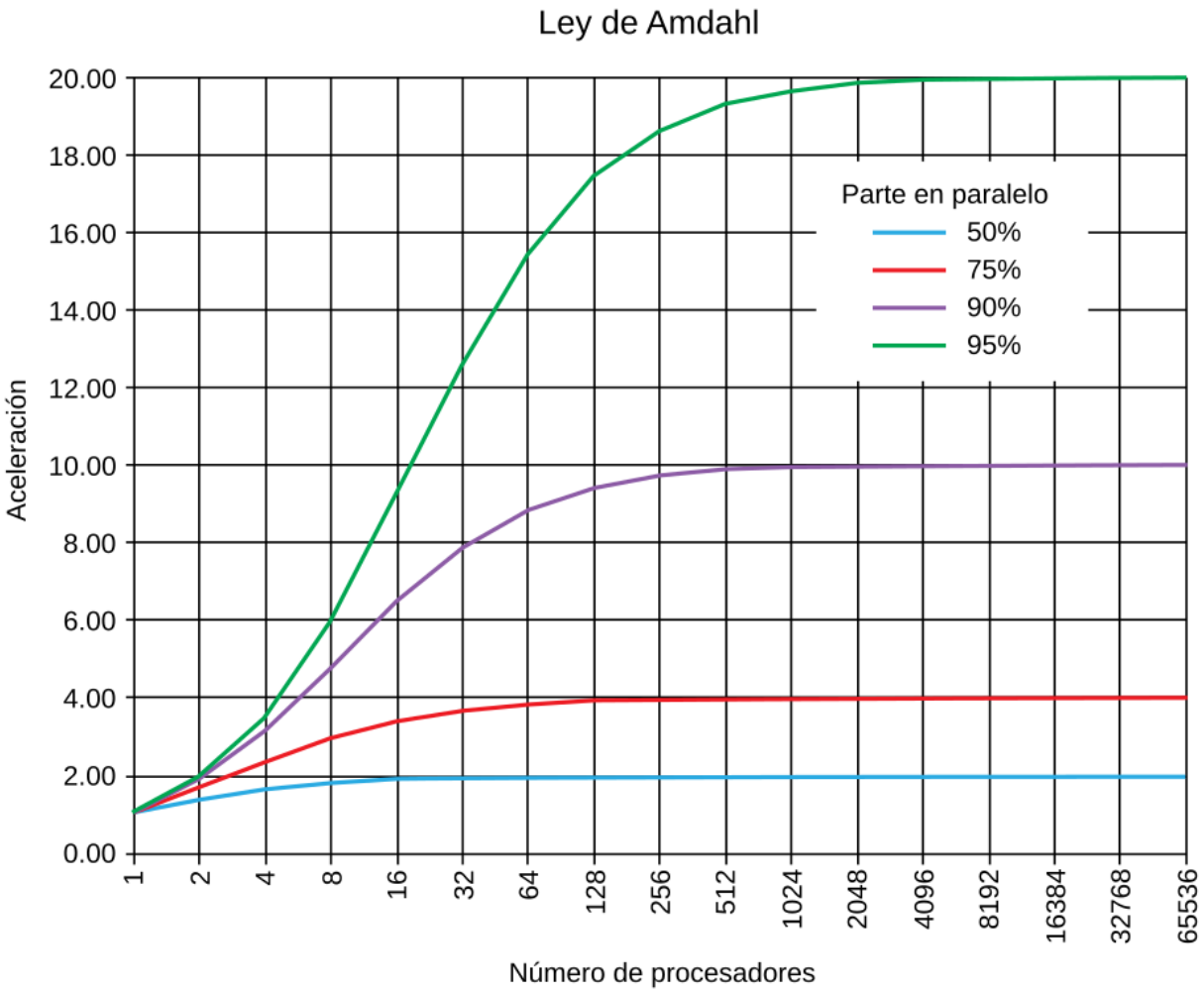
¿Hasta qué punto podemos paralelizar?

- N : Número de "trabajadores"
- P : Fracción paralelizable del código (entre 0 y 1)
- $\mathrm{Speedup}$: Métrica de aceleración

$$\mathrm{Speedup}(N) = \frac{1}{(1-P) + \frac{P}{N}}$$

- Eficiencia : $\frac{\mathrm{Speedup}(N)}{N}$

Ley de Amdahl



Conclusión

- Paralelizar require atención a los detalles.
- Existen infraestructuras y tecnologías.
- Necesidad del uso eficiente de los recursos disponibles.