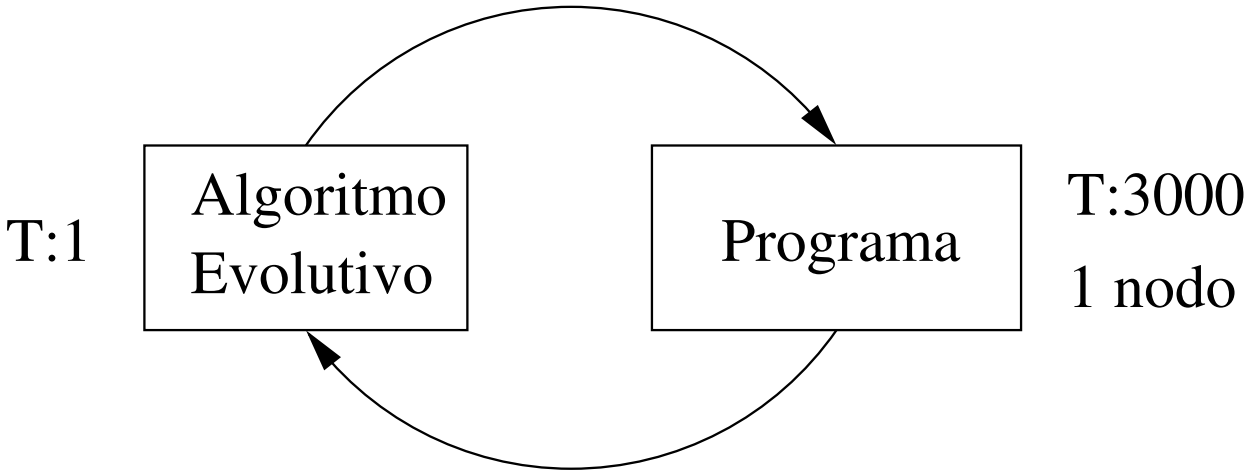
Realización de la Paralelización de Programas

Luis Gerardo de la Fraga

Correo-e: fraga@cs.cinvestav.mx
Departamento de Computación
Cinvestav Zacatenco

Agosto 22, 2012

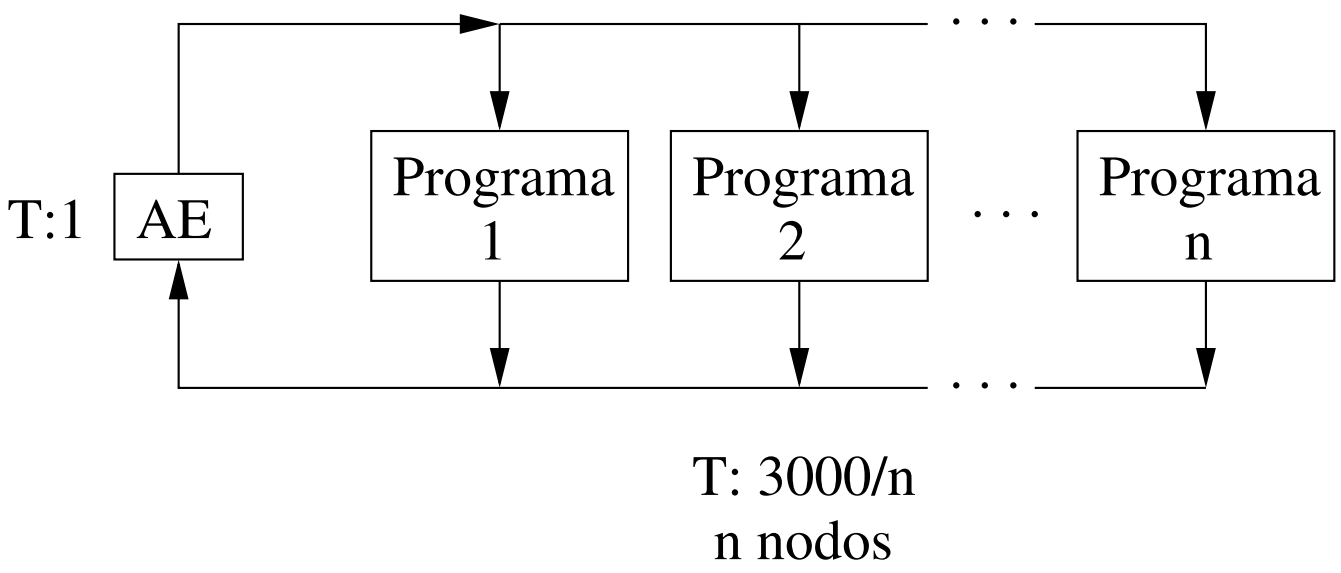
Contenido



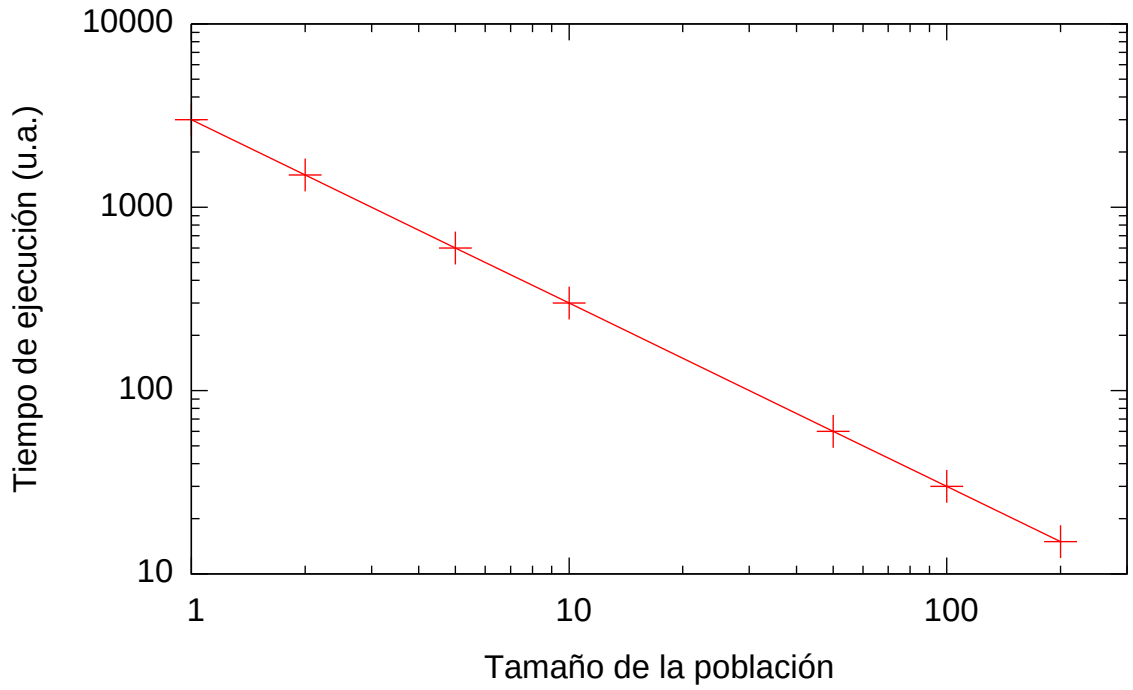
- El algoritmo evolutivo tiene tiempo de ejecución despreciable
- El tiempo de ejecución depende del *programa* que se debe ejecutar miles de veces (al menos 3,000).

¿Cómo reducir el tiempo de ejecución?

1. Ejecutar varias instancias del programa



n podría ser el tamaño de la población, esto es, el número de soluciones involucradas en una iteración del AE.



- En [1] se pudo ejecutar cada instancia del programa en una hebra del GPU
- Este problema es muy simple: es un problema de valor inicial.

Para ejecutar varias instancias se requiere el programa en un lenguaje de programación más amigable: perl, python, C o C++.

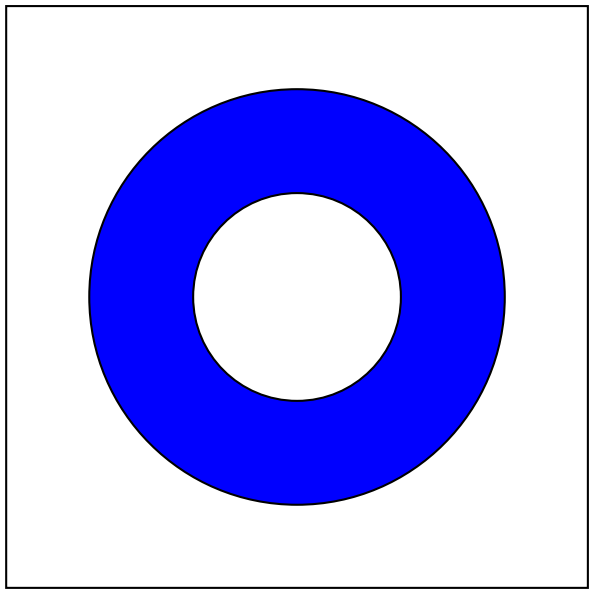
2. Cambiar de lenguaje de programación

- El cambio del programa de Matlab a C para maximizar el coeficiente de Lyapunov en un oscilador caótico [2] redujo el tiempo de ejecución de 1 día a unos pocos segundos.
- Este problema es también un problema de valor inicial en tres dimensiones.

3. Cambiar los algoritmos del programa

- El programa de ajuste de elipses es un problema de mínimos cuadrados que se resuelve como un problema generalizado de eigenvectores.
- Se puede resolver [3] como tres eigendescomposiciones de matrices de tamaño 3×3, que equivale a resolver tres ecuaciones cúbicas.

- El problema de encontrar los pixels dentro de un círculo es un problema de complejidad $O(n^2)$



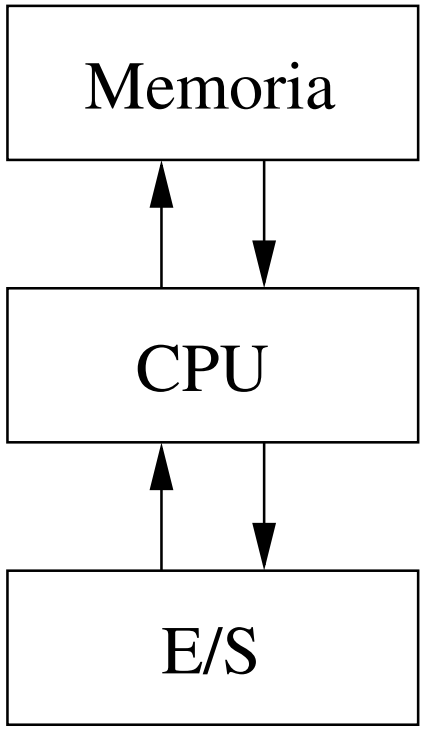
1: $c = N/2$
2: $r_2 = \text{radio} * \text{radio}$
3: **for** $i = 1 : N$ **do**
4: $\quad y = i - c$
5: $\quad y_2 = y * y$
6: $\quad$ **for** $j = 1 : N$ **do**
7: $\quad\quad x = j - c$
8: $\quad\quad x_2 = x * x$
9: $\quad\quad$ **if** $x_2 + y_2 <= r_2$ **then**
10: $\quad\quad\quad$ El pixel (i, j) está dentro del círculo

4. Ok, paralelizar el programa

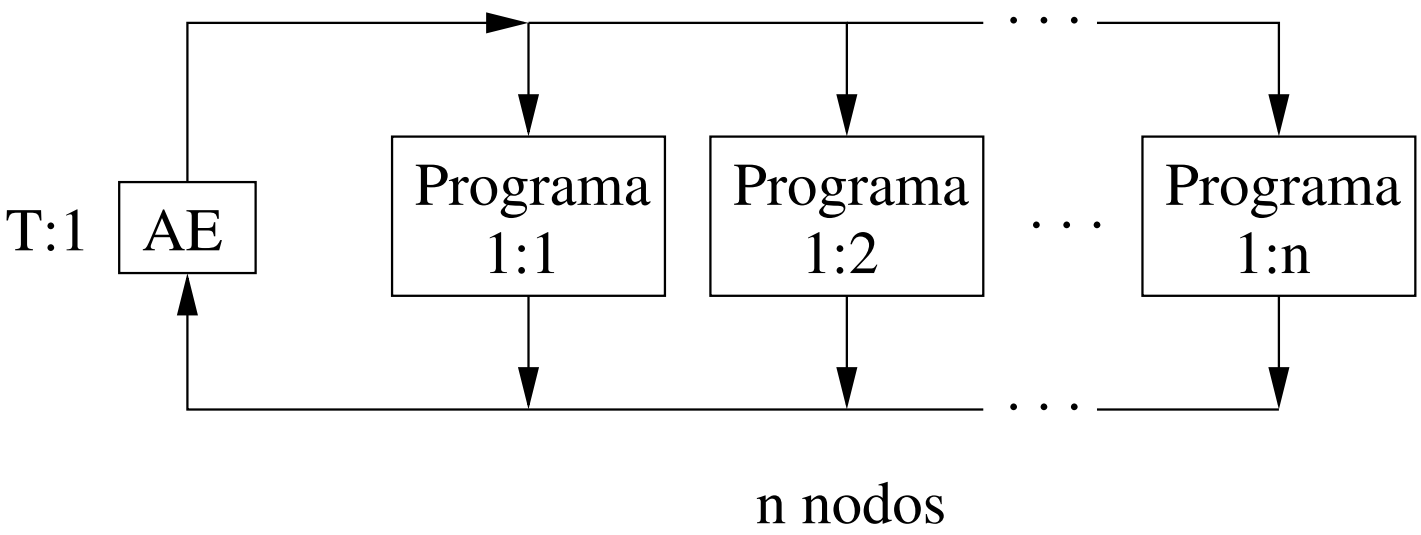
- Este problema es equivalente al de distribuir el problema en varios nodos.
- Distribuir: la instancia del problema es tan grande que no puede ejecutarse con los recursos de un nodo.

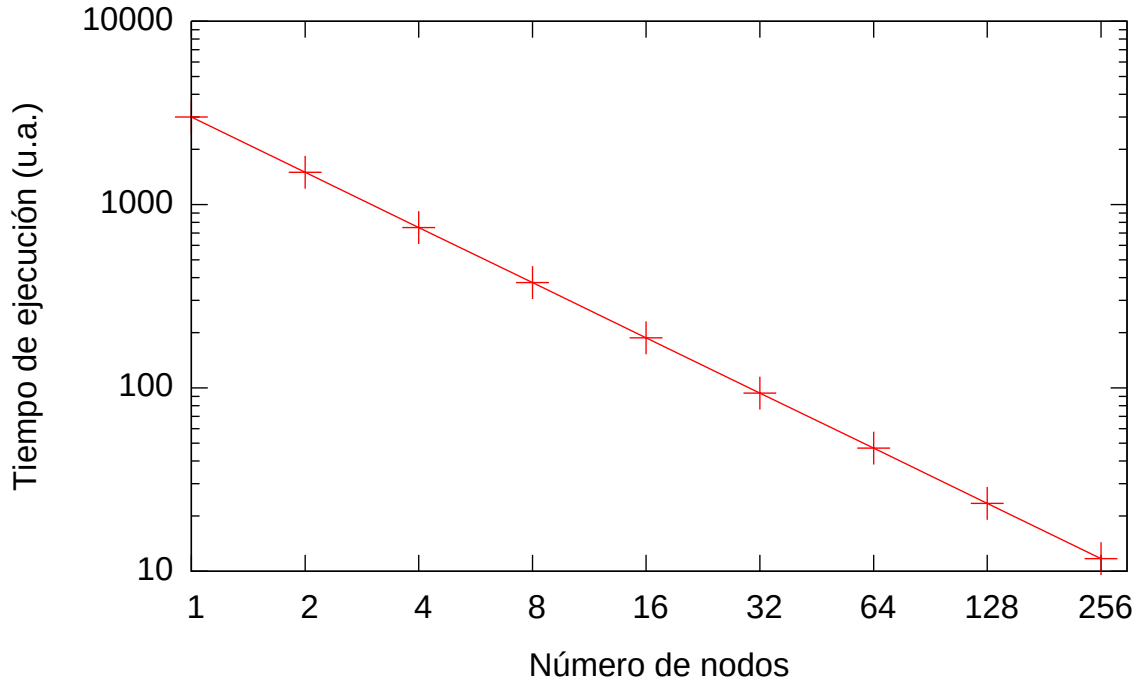
Paralelización

- Depende de la arquitectura de la computadora.



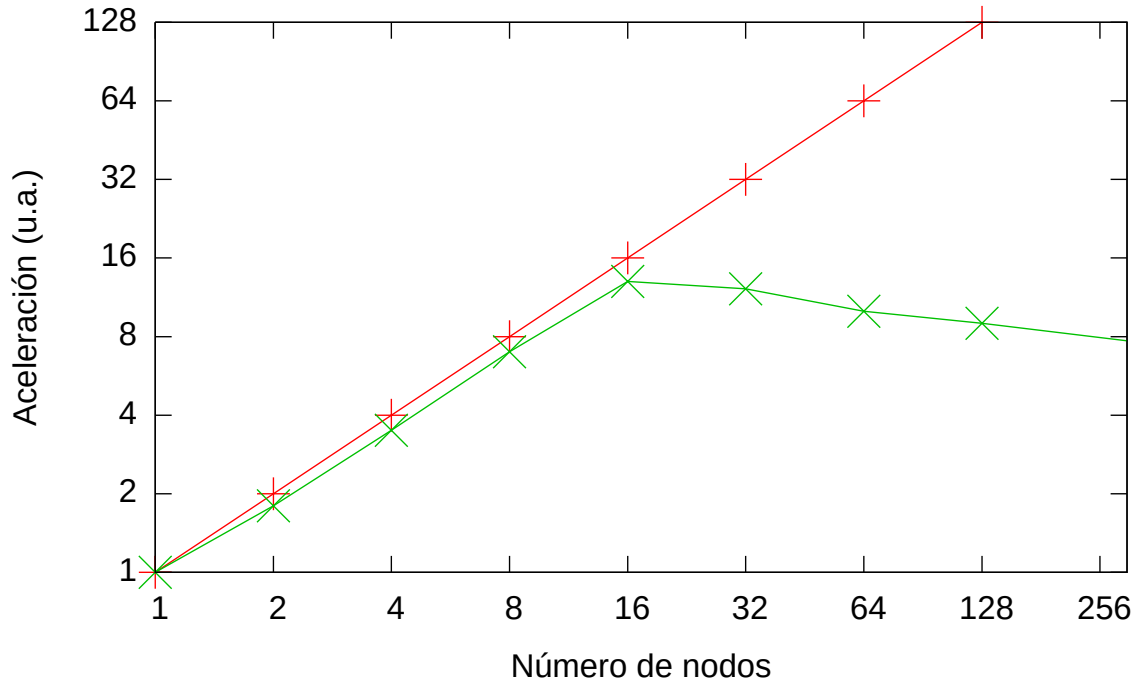
- A bajo nivel la paralelización se puede realizar si se cuenta con bibliotecas de análisis numérico:
  1. Invertir matrices
  2. Eigendescomposición
  3. Problemas de valor inicial
  4. Problemas de frontera
  5. Transformada de Fourier


T:1
AE
Programa
1:1
Programa
1:2
Programa
1:n
n nodos


10000
1000
100
10
Tiempo de ejecución (u.a.)
1
2
4
8
16
32
64
128
256
Número de nodos

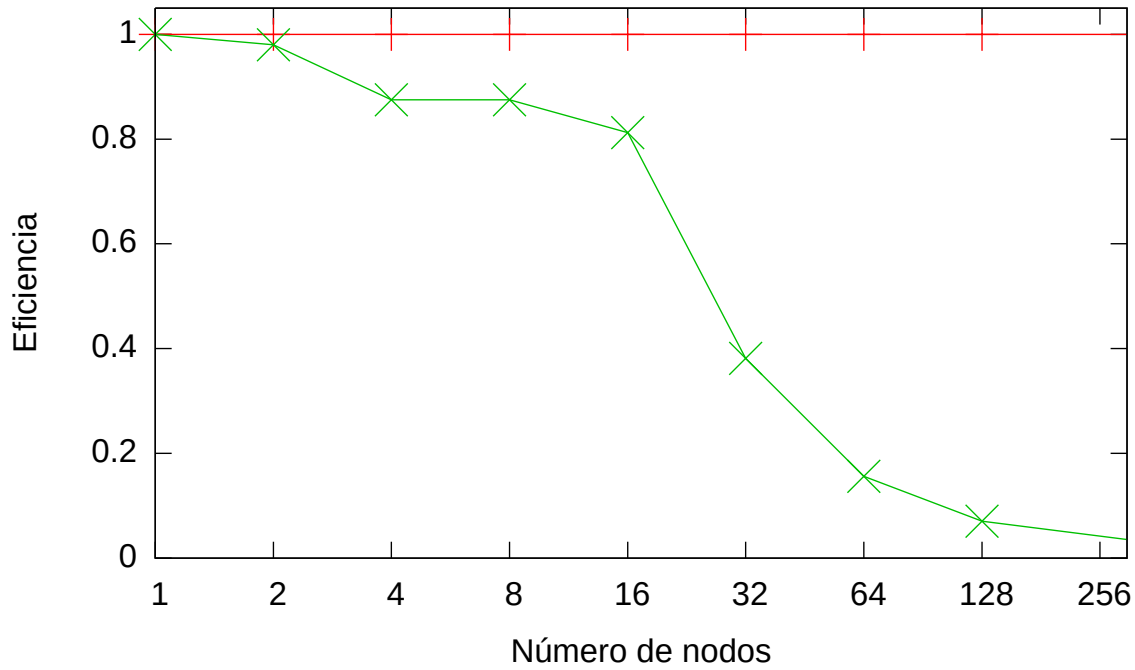
Si definimos:

$$\text{aceleración} = \frac{\text{Tiempo de ejecución del algoritmo secuencial}}{\text{Tiempo de ejecución del algoritmo paralelo}}$$



Si definimos:

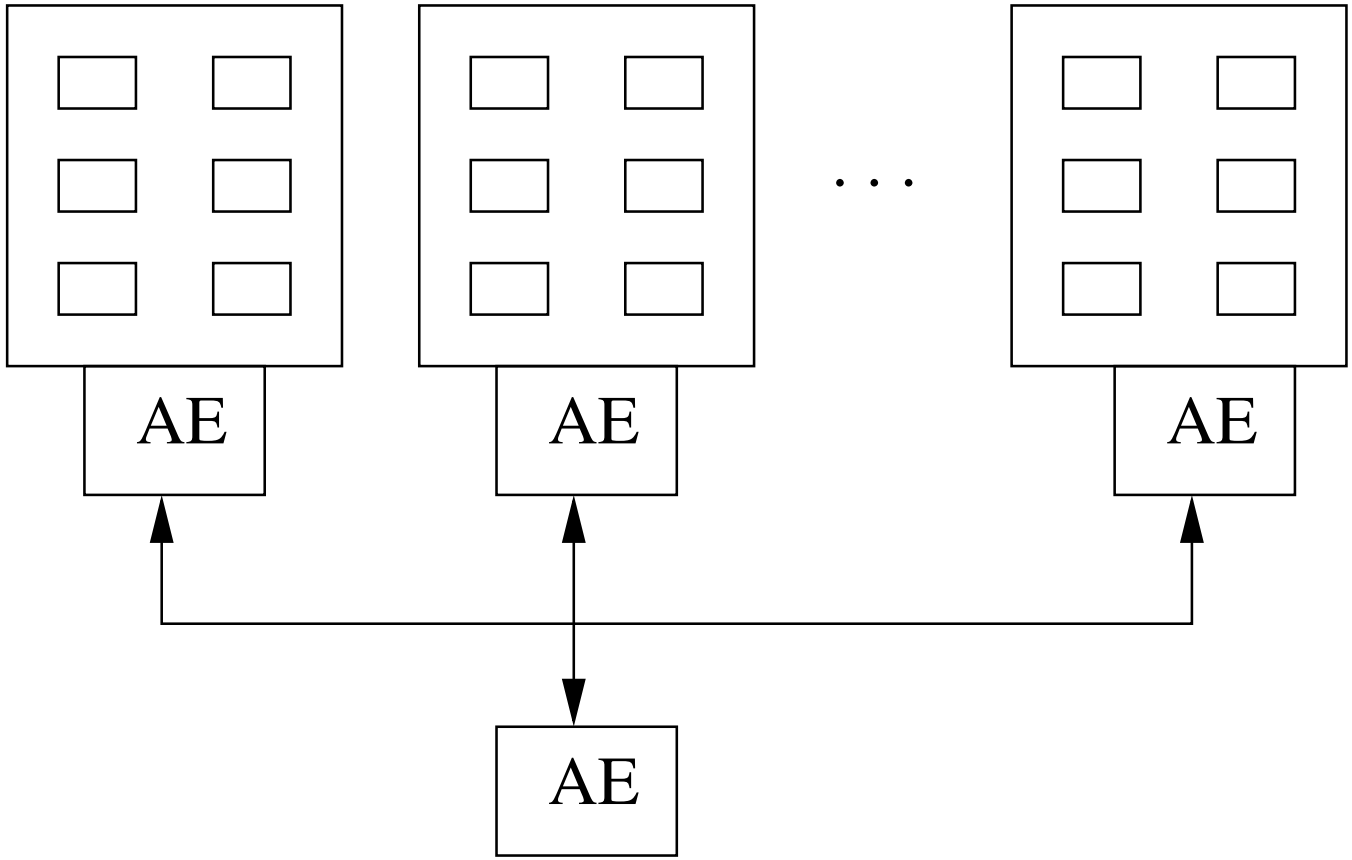
$$\text{eficiencia} = \frac{\text{Tiempo de ejecución del algoritmo secuencial}}{(\text{Tiempo de ejecución del algoritmo paralelo})(\text{no. nodos})}$$



Para paralelizar un programa

- Se debe contar con una versión secuencial del programa a paralelizar
- Se realiza el perfilado del programa secuencial para identificar las funciones (o métodos) que consuman el mayor tiempo de ejecución.
- Paralelizar esas partes del programa ó
- Usar una biblioteca ya paralelizada de métodos numéricos

¿Paralelización final?



Conclusiones (1/2)

- Para paralelizar un programa se necesita contar con un programa secuencial codificado en C o C++
- Se necesita conocer la arquitectura de la computadora
- ¿Se cuentan con bibliotecas numéricas ya paralelizadas?

Conclusiones (2/2)

- En una supercomputadora no se tiene una interfaz interactiva
- El programa paralelo se debe estar probado y funcionando
- En la supercomputadora se envia el programa a ejecutar y tiempo después (¿días?) se tiene el resultado
- Se puede realizar mucha investigación a nivel de la lógica –de la algorítmica– del programa.

¡Gracias!

Luis Enrique Ramírez Chávez.

Declaración de un algoritmo evolutivo en GPU's para la inferencia de modelos de redes reguladoras de genes.

Master's thesis, Departamento de Computación, Cinvestav, 2012.

Asesor Dr. Carlos Coello Coello, coasesor Dr. Eduardo Arturo Rodríguez Tello.

L.G. de la Fraga, E. Tlelo-Cuautle, V.H. Carbajal Gómez, and J.M. Muñoz Pacheco.

On maximizing positive lyapunov exponents in a chaotic oscillator with heuristics.

Revista Mexicana de Física, 58(3):274–281, June 2012.

Luis Gerardo de la Fraga and César Cruz Díaz.

Fitting an ellipse is equivalent to find the roots of a cubic equation.

In *2011 8th International Conference on Electrical Engineering, Computer Science and Automatic Control, Mérida, México*, pages 743–746, October 2011.