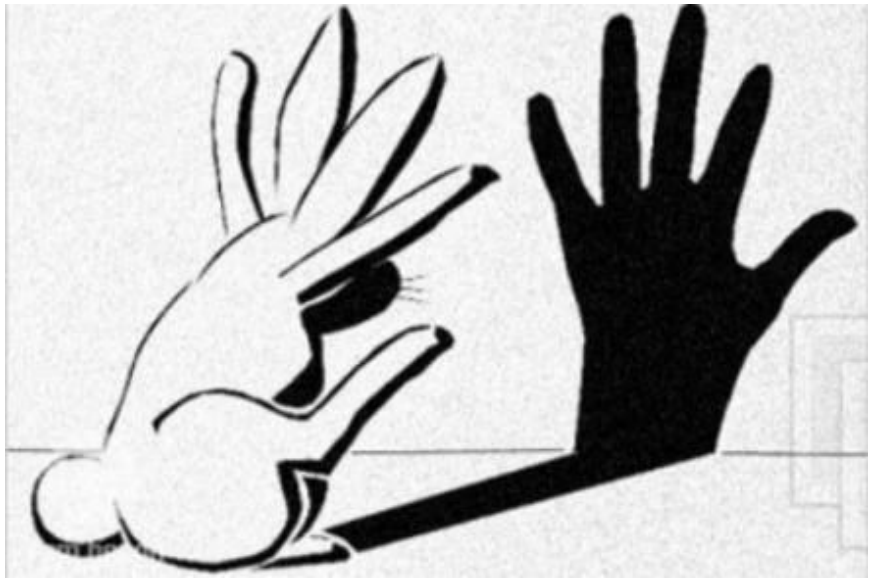
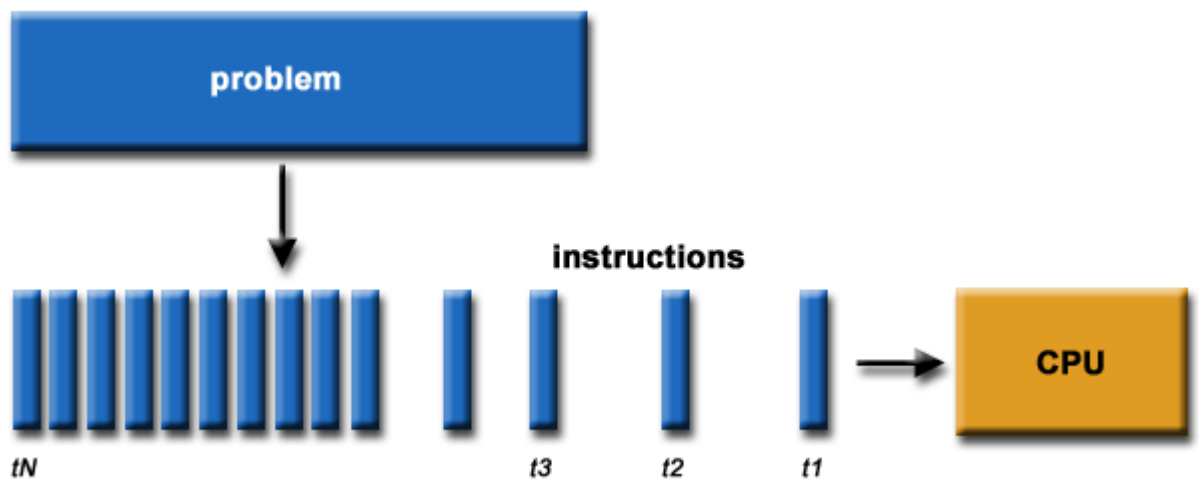


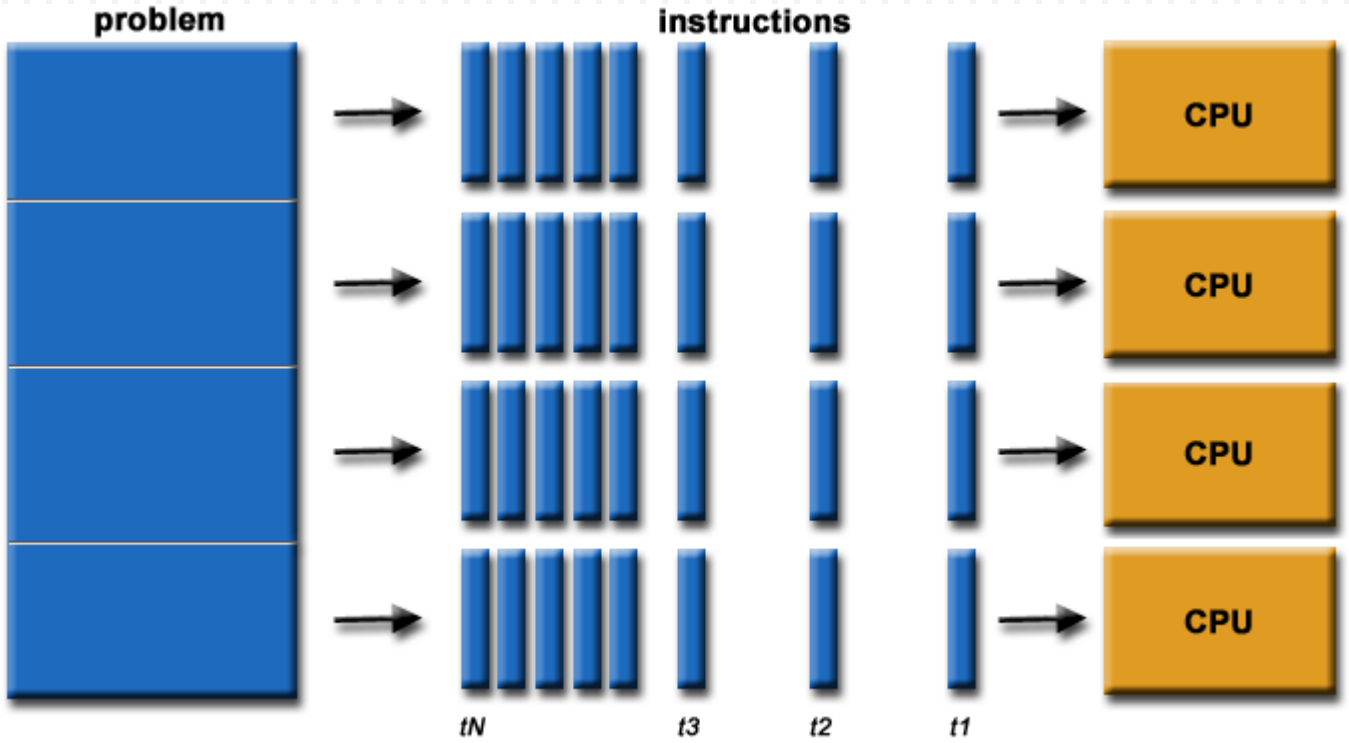
CÓMPUTO PARALELO

Netz Romero

Cómputo en serie



Cómputo en paralelo



Ejemplo

Realizar la siguiente expresión aritmética: `r=a*b+c*d`

- 1 Procesador

```
r1=a*b
r2=c*d
r=r1+r2
```

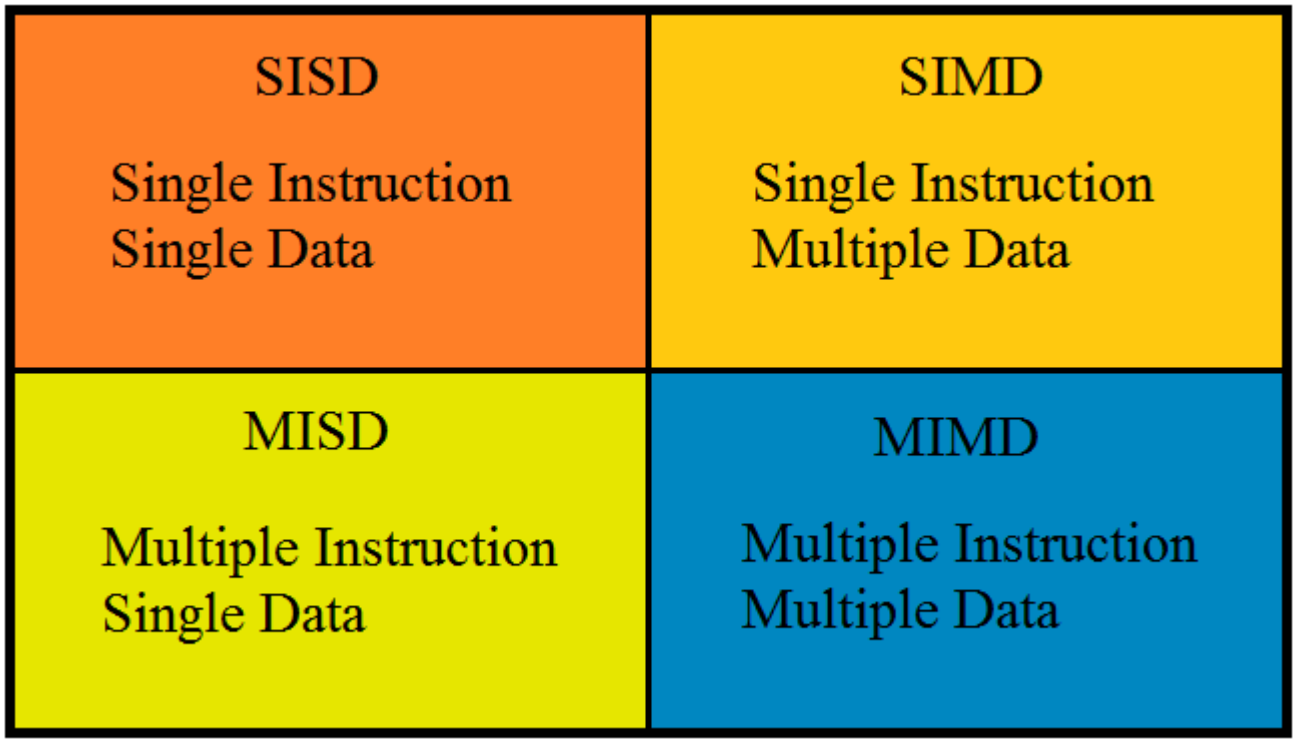
- 2 Procesadores

```
if(procesador==1)
    r1=a*b
if(procesador==2)
    r2=c*d
if(procesador==1)
    r=r1+r2
```

Cómputo Paralelo

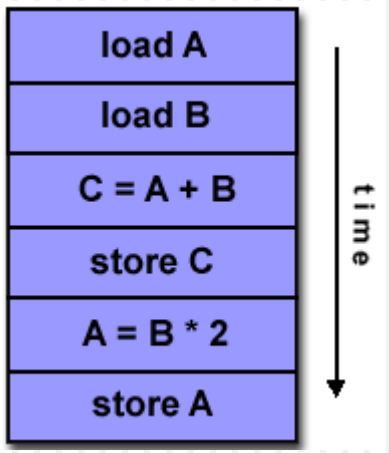
- La computación paralela es una forma de cómputo en la que muchas instrucciones se ejecutan simultáneamente [1]

La taxonomía de Flynn



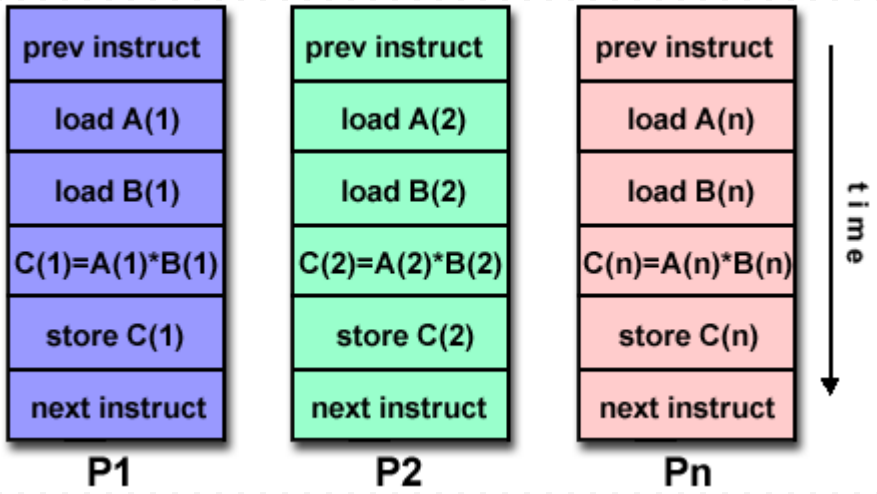
Single Instruction, Single Data (SISD)

- Es cómputo en serie, donde un procesador recibe una sola secuencia de instrucciones que opera una secuencia de datos.



Single Instruction, Multiple Data (SIMD)

- Es un tipo de cómputo paralelo, donde se tienen múltiples procesadores que ejecutan la misma secuencia de instrucciones, pero en diferentes datos.



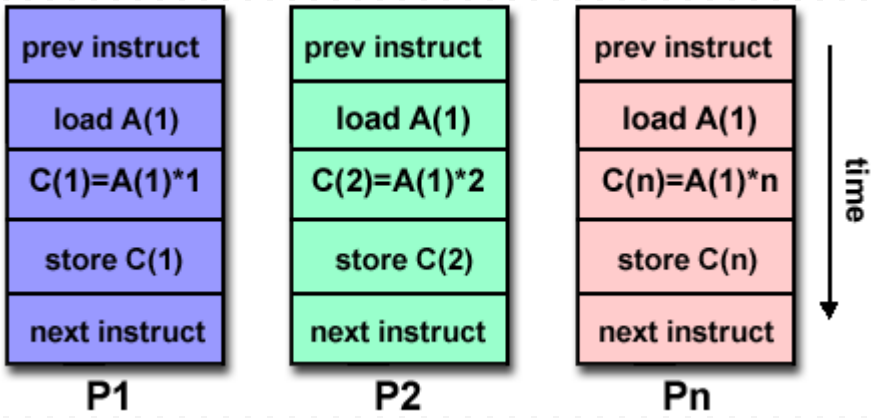
- Como ejemplo, se suman las matrices A y B con cuatro procesadores.

$$A=\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad B=\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \quad C=\begin{bmatrix} C_{11}=A_{11}+B_{11} & C_{12}=A_{12}+B_{12} \\ C_{21}=A_{21}+B_{21} & C_{22}=A_{22}+B_{22} \end{bmatrix}$$

$C_{11} = A_{11} + B_{11}$	$C_{12} = A_{12} + B_{12}$	$C_{21} = A_{21} + B_{21}$	$C_{22} = A_{22} + B_{22}$
Procesador 1	Procesador 2	Procesador 3	Procesador 4

Multiple Instruction, Single Data (MISD)

- Es un tipo de cómputo paralelo, con diferentes secuencias de instrucciones para cada procesador y con el mismo dato ó compartiendo memoria común.

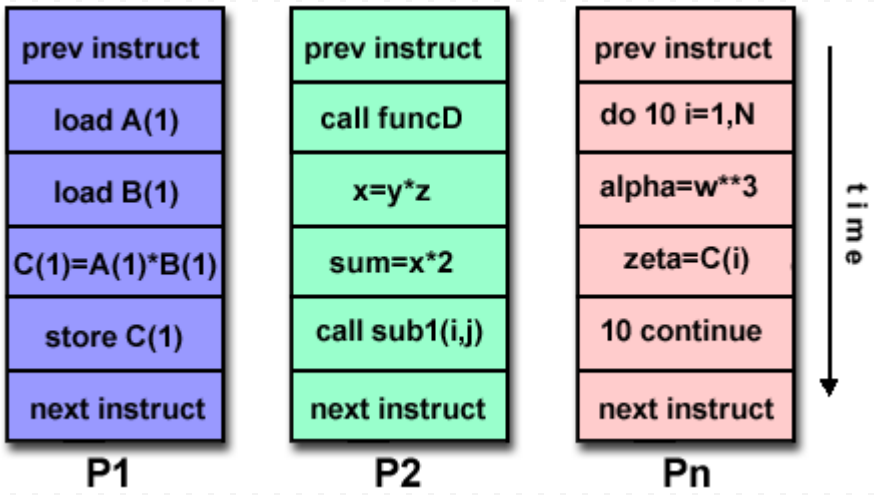


- Como ejemplo, sumar, restar, multiplicar y dividir dos números en a y b con cuatro procesadores.

S = a+b	R = a-b	M = a*b	D = a/b
Procesador 1	Procesador 2	Procesador 3	Procesador 4

Multiple Instruction, Multiple Data (MIMD)

- Es un tipo de cómputo paralelo, donde cada procesador puede ejecutar su propia serie de instrucciones y tener sus propios datos.

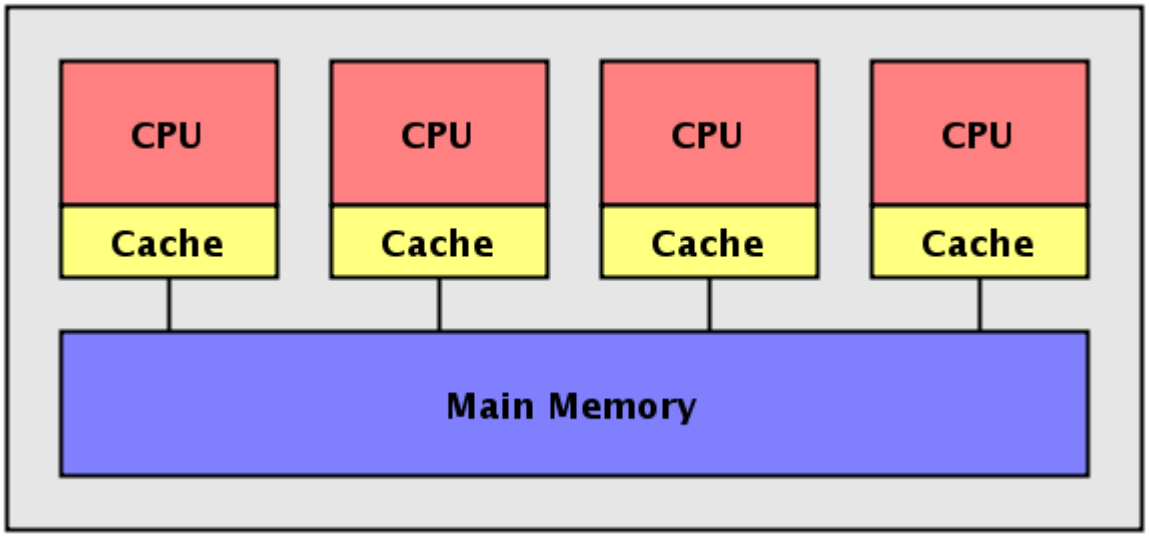


- Como ejemplo, se desea calcular diferentes operaciones con diferentes datos en cuatro procesadores.

S = a+b	R = c-d	M = e*f	D = g/h
Procesador 1	Procesador 2	Procesador 3	Procesador 4

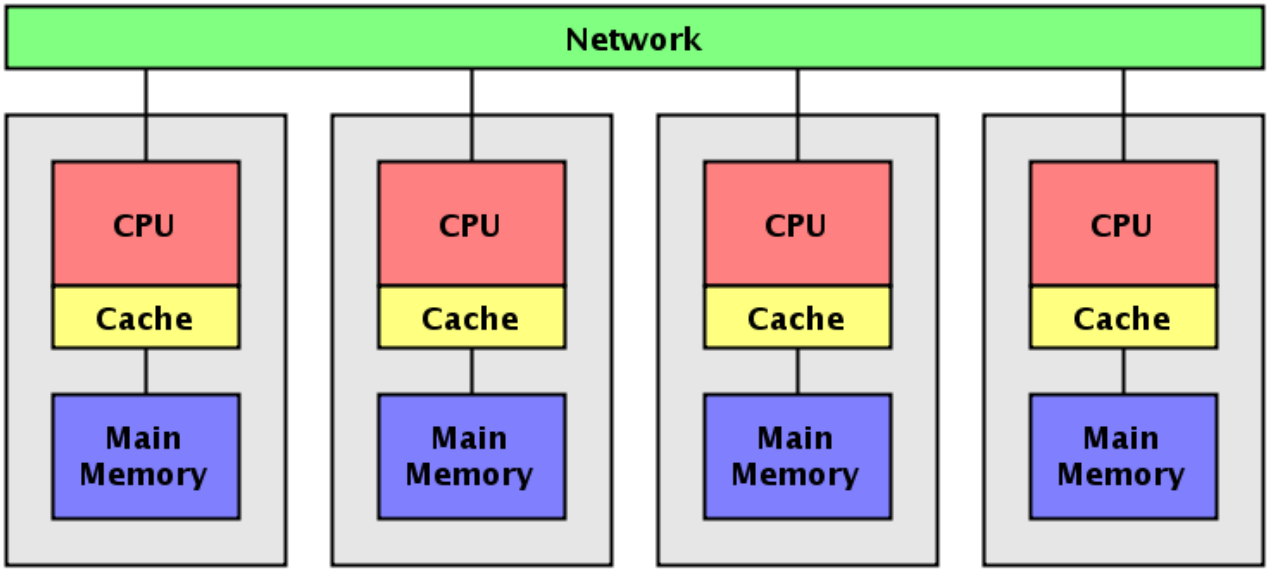
Memoria compartida

- La memoria compartida permite crear segmentos de memoria para que pueda ser accedida por múltiples procesos.



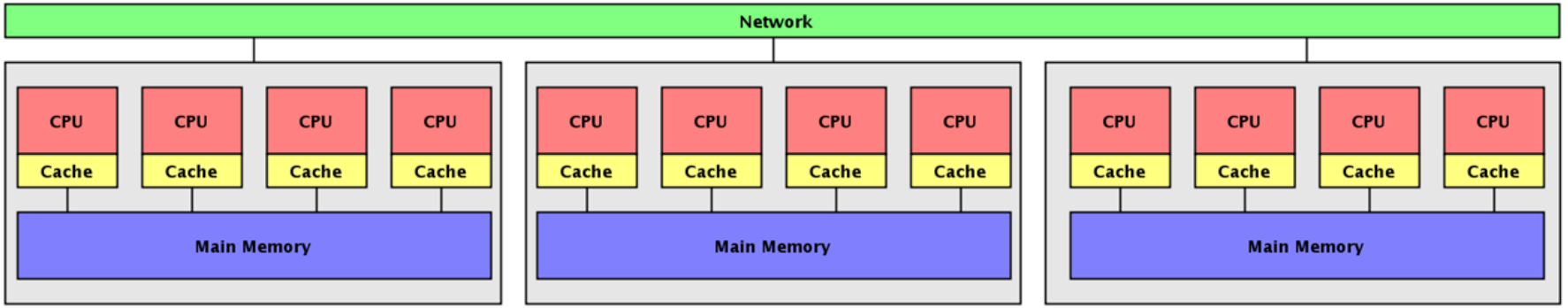
Memoria distribuida

- La memoria distribuida se refiere a un sistema con múltiples procesadores donde cada uno de ellos tiene su propia memoria privada.



Memoria distribuida compartida

- Este tipo de memoria, donde los procesadores de cada nodo tienen acceso a la memoria compartida y ésta corresponde a la memoria privada de cada nodo dentro de un cluster.



Tipos de MIMD

- MIMD con memoria compartida

  Cada procesador tiene acceso a toda la memoria utilizando un ducto en común.

- MIMD con memoria distribuida

  Cada procesador tiene una memoria local y solo pueden compartir la información con el mando de mensajes.

MIMD con memoria compartida

- Ventajas

  Son fáciles de programar

- Desventajas

  Acceso simultáneo

  Poca escalabilidad

Las computadoras MIMD con memoria compartida son conocidas como Sistemas de Multiprocesamiento Simétrico.

MIMD con memoria distribuida

- Ventajas

  Son fáciles de escalar

- Desventajas

  El acceso remoto puede ser lento

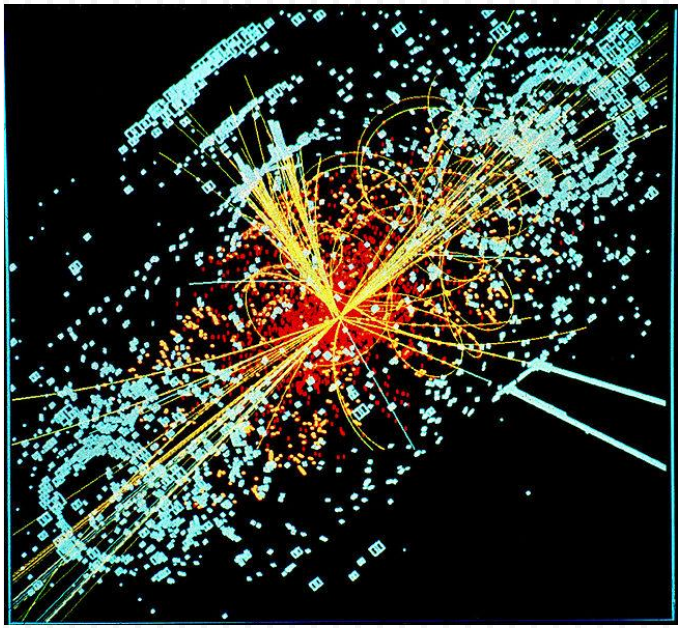
Las computadoras MIMD con memoria distribuida son conocidas como Sistemas de Procesamiento en Paralelo Masivo.

Aplicaciones de cómputo paralelo

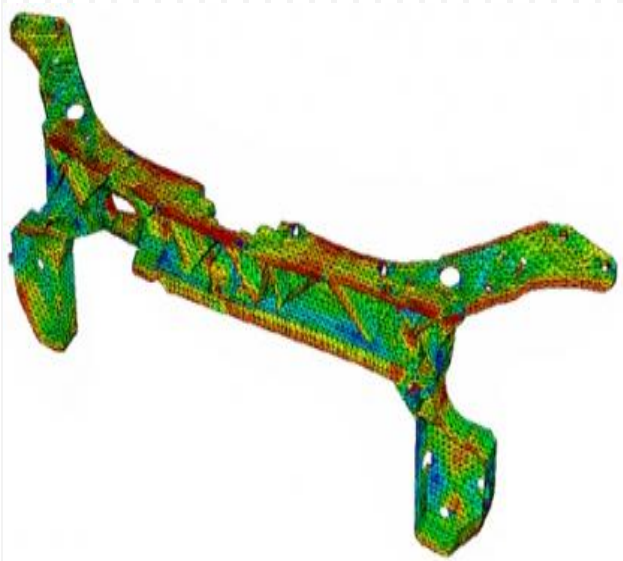
En ciencia e ingeniería

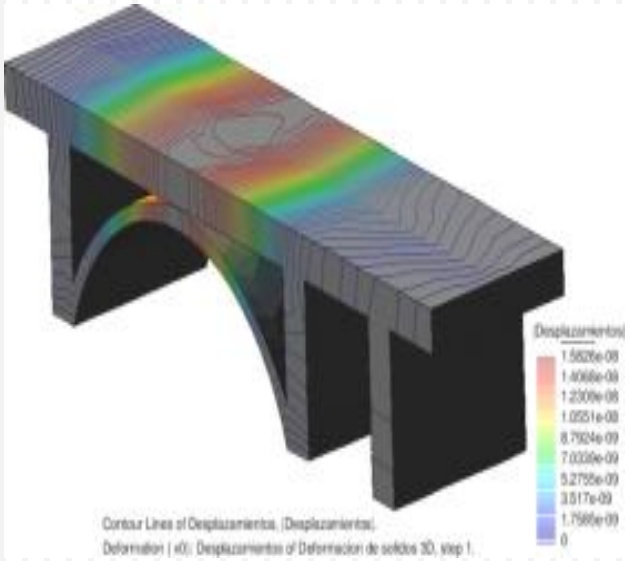
- Sistemas de información geográfica
- Física de partículas, ciencias nucleares, astronomía
- Biotecnología, genética
- Mecánica de materiales, modelado de estructuras
- Ciencias de la computación, matemáticas

Física



Materiales y estructuras



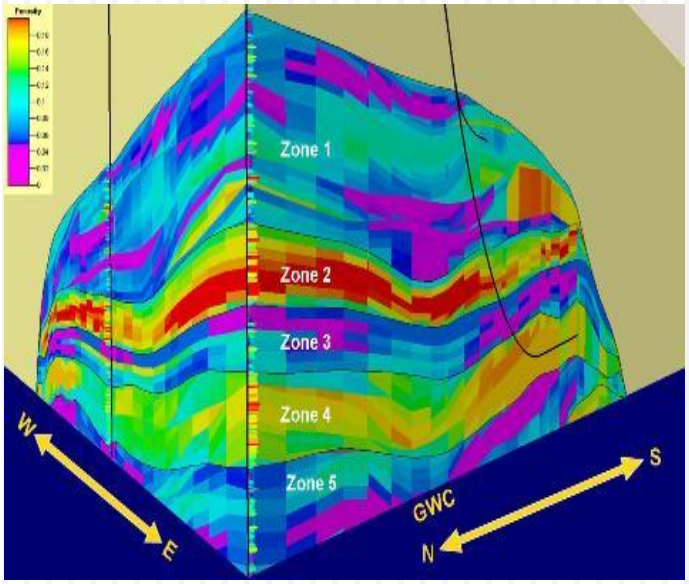


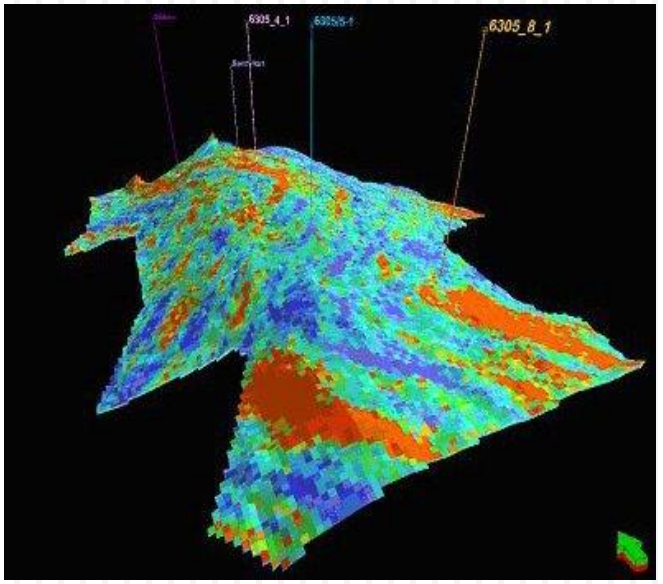
Aplicaciones de cómputo paralelo

Aplicaciones industriales y comerciales

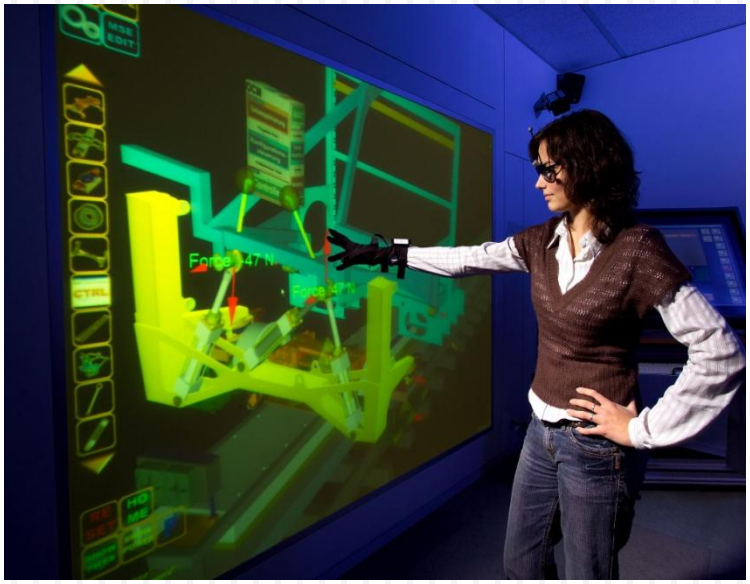
- Bases de datos, minería de datos
- Explotación petrolera
- Sistemas distribuidos web
- Modelos económico – financiero
- Realidad virtual, videojuegos

Explotación petrolera





Realidad virtual y videojuegos



¿Porqué utilizar cómputo paralelo?

- Aumentar la velocidad
- Procesa una gran cantidad de datos
- Resuelve problemas en tiempo real
- Resuelve problemas complejos

```
1   #include <pthread.h>
2   #include <stdio.h>
3   #define NUM_THREADS 5
4   void *PrintHello(void *threadid)
5   {
6    long tid;
7    tid = (long)threadid;
8    printf("Hello World! It.s me, thread # %ld!\n", tid);
9    pthread_exit(NULL);
10  }
11  int main (int argc, char *argv[])
12  {
13   pthread_t threads[NUM_THREADS];
14   int rc;
15   long t;
16   for(t=0; t<NUM_THREADS; t++){
17    printf("In main: creating thread %ld\n", t);
18    rc = pthread_create(&threads[t], NULL, PrintHello, (void *)t);
19    if (rc){
20       printf("ERROR; return code from pthread_create() is %d\n", rc);
21       return(-1);
22     }
23   }
24   pthread_exit(NULL);
25  }
```

```
1  public class Runner {
2    public static void main(String[] args){
3      HelloRunner r = new HelloRunner();
4      Thread t1 = new Thread(r);
5      t1.start();
6           for(int i = 0; i < 4; i++){
7               System.out.println("Yo soy el main, tiempo: " + i);
8               try {    Thread.sleep(1000);
9               } catch (InterruptedException ex) { ex.printStackTrace(); }
10          }
11     System.out.println("Fin del main");
12   }
13 }
14 class HelloRunner implements Runnable{
15     int i;
16     public void run(){
17         i = 0;
18         while(true){
19             System.out.println("Hola: "+ i++);
20             try {    Thread.sleep(100);
21             } catch (InterruptedException ex) { ex.printStackTrace(); }
22             if(i==40) break;
23         }
24         System.out.println("Fin del hilo");
25     }
26 }
```

- Para compilar en C

```
gcc printHello.c -lpthread -o printHello
./printHello
```

- Para compilar en Java

```
javac Runner.java
java Runner
```

Referencias

- Taxonomía de Flynn, http://132.248.9.195/pd2006/0606805/A4.pdf
- Aplicaciones, https://computing.llnl.gov/tutorials/parallel_comp/
- Tipos de memoria, http://www.cs.rit.edu/~ark/lectures/pj04/notes.shtml