

## Paralelismo y Concurrencia en Sistemas Programación en Multicomputadoras



Dr. Alejandro J. García  
e-mail: agarcia@cs.uns.edu.ar  
http://cs.uns.edu.ar/~ajg



Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur  
Bahía Blanca - Argentina

### Programar en paralelo

**Multiprocessor:** el programador ve su programa como una **colección** de **procesos** accediendo a un conjunto de **variables compartidas**.

**Multicomputer:** el programador ve su programa como una **colección** de **procesos** (Ps), cada uno de ellos con sus propias variables locales, y la habilidad de **enviar** y **recibir mensajes** de otros Ps.

- Para **programar efectivamente** en un entorno "multicomputadora" hay que tener cierto **conocimiento** de su **arquitectura**.
- En general, **tomar un programa eficiente para multiprocessor y ejecutarlo en una arquitectura multicomputer lleva a una baja performance**.

Paralelismo y Concurrencia en Sistemas Dr. Alejandro J. García

2

### Multi. de matrices con Memoria compartida

```
PROGRAM mat;
var i,j: integer; A,B,P:matrix; {compartidas}
  Procedure mult(i,j:integer);
  var s: real; k: integer;
  begin s:=0;
    for k:=1 to max do s:= s + A[i,k] * B[k,j];
    P[i,j]:= s;
  end;
begin
FORALL i:=1 TO filas DO
  FORALL j:=1 TO columnas DO mult(i,j);
end.
```

Paralelismo y Concurrencia en Sistemas

Dr. Alejandro J. García

3

### Multiplicación de matrices (asumiendo $N \times N$ )

**Arq. Multiprocessor:**

- necesito  $N^2$  procesadores y uso  $3 \cdot N^2$  celdas mem.

**Arq. Multicomputer:**

- no hay acceso a memoria compartida
  - no hay "infinitos" procesadores
- Si se usa el **mismo algoritmo (empeoran las cosas)**:
- cada proceso necesita 1 fila y 1 columna
  - si copia fila y columna usaría  $N^2 \times 2N = 2N^3$  celdas
  - empeora el tiempo de ejecución
- si cada Pc calcula una fila, necesita las dos matrices, entonces igual hay  $N \times 2N^2 = 2N^3$  celdas

Paralelismo y Concurrencia en Sistemas

Dr. Alejandro J. García

4

### Paralelismo de Datos

- En el **paralelismo de datos** diferentes partes de una misma base de datos son procesados en paralelo.
- En **arg. multiprocesadores** la base de datos está en memoria compartida, lo cual facilita las cosas.
- En **arg. Multicomputadoras** la memoria distribuida requiere de un tipo de paralelismo llamado "**particionamiento de datos**".
- La base de datos es **particionada** y cada parte almacenada en cada módulo de memoria.
  - Cada proceso opera sobre su **memoria local** y accede a datos remotos **vía mensajes**.

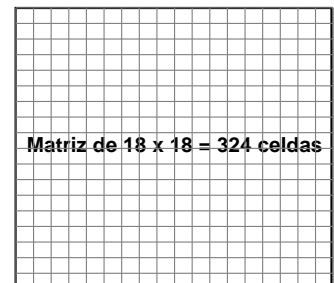
Paralelismo y Concurrencia en Sistemas

Dr. Alejandro J. García

5

### Multiplicación de matrices en Multicomputer

- Un procesador por celda puede ser impracticable



Paralelismo y Concurrencia en Sistemas

Dr. Alejandro J. García

6

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:  
"Paralelismo y Concurrencia en Sistemas. Notas de Clase". Alejandro J. García. Universidad Nacional del Sur. (c) 2002-2010.

### Multiplicación de matrices en Multicomputer

- Un procesador por celda puede ser impracticable
- Solución:** Partición y distribución de los datos

Parte 1-1 al Pc 0	Parte 1-2 al Pc 1	Parte 1-3 al Pc 2
Parte 2-1 al Pc 3	Parte 2-2 al Pc 4	Parte 2-3 al Pc 5
Parte 3-1 al Pc 6	Parte 3-2 al Pc 7	Parte 3-3 al Pc 8

Paralelismo y Concurrencia en Sistemas Dr. Alejandro J. García 7

### Multiplicación de matrices en Multicomputer

- Un procesador por celda puede ser impracticable
- Solución:** Partición y distribución de los datos
- Cada procesador necesita datos de sus vecinos
- Arquitectura: TORUS

Parte 1-1 al Pc 0	Parte 1-2 al Pc 1	Parte 1-3 al Pc 2
Parte 2-1 al Pc 3	Parte 2-2 al Pc 4	Parte 2-3 al Pc 5
Parte 3-1 al Pc 6	Parte 3-2 al Pc 7	Parte 3-3 al Pc 8

Paralelismo y Concurrencia en Sistemas Dr. Alejandro J. García 8

### Ejemplo:

- matriz de 3x3 (sin agrupamiento):
- Cada procesador tendrá una parte de A y de B
- Y calculará su parte de C con los datos que recibe de su fila y columna
- Pc1-1 calcula C11 = A11xB11 + A12xB21 + A13xB31

Pc 1-1 A11 B11 C11	Pc 1-2 A12 B12 C12	Pc 1-3 A13 B13 C13
Pc 2-1 A21 B21 C21	Pc 2-2 A22 B22 C22	Pc 2-3 A23 B23 C23
Pc 3-1 A31 B31 C31	Pc 3-2 A32 B32 C32	Pc 3-3 A33 B33 C33

Paralelismo y Concurrencia en Sistemas Dr. Alejandro J. García 9

### Ejemplo:

- El problema es que inicialmente, el Pc 1-2 tiene que esperar.
- Además, necesita guardar algunos datos para cuando lleguen los otros.
- Todos los que no están en la diagonal tienen el mismo problema.
- Problema en C12 = A11xB12 + A12xB22 + A13xB32

Pc 1-1 A11 B11 C11	Pc 1-2 A12 B12 C12	Pc 1-3 A13 B13 C13
Pc 2-1 A21 B21 C21	Pc 2-2 A22 B22 C22	Pc 2-3 A23 B23 C23
Pc 3-1 A31 B31 C31	Pc 3-2 A32 B32 C32	Pc 3-3 A33 B33 C33

Paralelismo y Concurrencia en Sistemas Dr. Alejandro J. García 10

### Solución:

- Desplazar datos inicialmente:
- La fila k está rotada a izquierda k-1 pasos, y la columna para arriba k-1 pasos
- Cada Pc calculará su parte de C con los datos que recibe de su fila y columna
- Pc1-1 calcula C11 = A11xB11 + A12xB21 + A13xB31
- Pc1-2 calcula C12 = A11xB12 + A12xB22 + A13xB32

Pc 1-1 A11 B11 C11	Pc 1-2 A12 B22 C12	Pc 1-3 A13 B33 C13
Pc 2-1 A22 B21 C21	Pc 2-2 A23 B32 C22	Pc 2-3 A21 B13 C23
Pc 3-1 A33 B31 C31	Pc 3-2 A31 B12 C32	Pc 3-3 A32 B23 C33

Paralelismo y Concurrencia en Sistemas Dr. Alejandro J. García 11

### Multi. de matrices con Memoria distribuida

```

Procedure mult(i,j;integer, vA, vB;real; VAR Resul;real);
var C: real; k: integer;
begin
  C:=0;
  for k:=1 to max do
    begin
      C:= C + vA*vB;
      Enviar vA al canal de la izquierda
      Enviar vB al canal de arriba
      Leer el nuevo vA del canal de la fila
      Leer el nuevo vB del canal de la columna
    end;
  Resul:=C;
end;
    
```

Paralelismo y Concurrencia en Sistemas Dr. Alejandro J. García 12

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:  
 "Paralelismo y Concurrencia en Sistemas. Notas de Clase". Alejandro J. García. Universidad Nacional del Sur. (c) 2002-2010.

**Paso 1:**

- Cada Pc comienza calculando su parte de C con los datos que tiene.

Pc 1-1 A11 B11	Pc 1-2 A12 B22	Pc 1-3 A13 B33
Pc 2-1 A22 B21	Pc 2-2 A23 B32	Pc 2-3 A21 B13
Pc 3-1 A33 B31	Pc 3-2 A31 B12	Pc 3-3 A32 B23

- Pc1-1 calcula  $C_{aux} = A_{11} \times B_{11}$
- Pc1-2 calcula  $C_{aux} = A_{12} \times B_{22}$

Paralelismo y Concurrencia en Sistemas Dr. Alejandro J. García 13

**Paso 1:**

- Cada Pc tiene dos canales cA y cB
- Cada Pc envía su parte de A al Pc de la izquierda y su parte de B al Pc de arriba.
- y por lo tanto recibe dos elementos

Pc 1-1 A11 B11	Pc 1-2 A12 B22	Pc 1-3 A13 B33
Pc 2-1 A22 B21	Pc 2-2 A23 B32	Pc 2-3 A21 B13
Pc 3-1 A33 B31	Pc 3-2 A31 B12	Pc 3-3 A32 B23

- Pc1-1 calcula  $C_{aux} = A_{11} \times B_{11}$
- Pc1-2 calcula  $C_{aux} = A_{12} \times B_{22}$

Paralelismo y Concurrencia en Sistemas Dr. Alejandro J. García 14

**Paso 2:**

- Cada Pc tiene dos canales cA y cB
- Cada Pc envía su parte de A al Pc de la izquierda y su parte de B al Pc de arriba.
- y por lo tanto recibe dos elementos

Pc 1-1 A12 B21	Pc 1-2 A13 B32	Pc 1-3 A11 B13
Pc 2-1 A23 B31	Pc 2-2 A21 B12	Pc 2-3 A22 B23
Pc 3-1 A31 B11	Pc 3-2 A32 B22	Pc 3-3 A33 B33

- Pc1-1 envía A11 y B11 y recibe A12 y B21
- Pc1-2 envía A12 y B22 y recibe A13 y B32

Paralelismo y Concurrencia en Sistemas Dr. Alejandro J. García 15

**Paso 3:**

- Cada Pc continúa el cálculo con los dos nuevos elementos recibidos

Pc 1-1 A12 B21	Pc 1-2 A13 B32	Pc 1-3 A11 B13
Pc 2-1 A23 B31	Pc 2-2 A21 B12	Pc 2-3 A22 B23
Pc 3-1 A31 B11	Pc 3-2 A32 B22	Pc 3-3 A33 B33

- Pc1-1 calcula  $C_{aux} = C_{aux} + A_{12} \times B_{21}$
- Pc1-2 calcula  $C_{aux} = C_{aux} + A_{13} \times B_{32}$

Paralelismo y Concurrencia en Sistemas Dr. Alejandro J. García 16

**Tipos/categorías/técnicas de paralelismo:**

- Relaxed parallelism.
- Paralelismo de datos.
- Non-relaxed parallelism
- Pipeline Parallelism
- Paralelismo sincrónico
- Particionamiento de datos:**
  - La base de datos es particionada y cada parte almacenada en cada módulo de memoria.
  - Cada proceso opera sobre su memoria local y accede a datos remotos vía mensajes.

Paralelismo y Concurrencia en Sistemas Dr. Alejandro J. García 17

**Bibliografía de referencia**

- The Art of Parallel Programming. Bruce P. Lester. 1993. Prentice Hall capítulos 7, 8 y 9

Paralelismo y Concurrencia en Sistemas Dr. Alejandro J. García 18

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:  
 "Paralelismo y Concurrencia en Sistemas. Notas de Clase". Alejandro J. García. Universidad Nacional del Sur. (c) 2002-2010.