

# Computación de Alta Performance

## Práctico 1

- Compare los sistemas monoprocesadores con los multiprocesadores (o paralelos) en los siguientes aspectos:
  - Facilidad de programación.
  - Necesidad de comunicaciones y sincronizaciones.
  - Evaluación de la performance.
- Que ventajas considera que tiene la programación distribuida sobre la programación paralela? (y vice versa). Ejemplifique con dos aplicaciones, una para cada caso.
- Considere una aplicación de procesamiento de imágenes que consiste de doce procesos: un proceso de entrada, un proceso de salida y diez procesos de cómputo. La tarea de entrada tiene que completarse antes que inicien las tareas de procesamiento. Similarmente, las tareas de procesamiento tienen que completarse antes de que inicie la tarea de salida. Cada una de las 12 tareas toma una unidad de tiempo. Existe un solo dispositivo de entrada y uno solo de salida.
  - ¿Cuál es el máximo speedup que se puede lograr con este problema cuando se utilizan dos procesadores?
  - ¿Cuál es la cota superior del speedup que se logra con esta aplicación cuando se utiliza una computadora paralela?
  - ¿Cuál es el menor número de procesadores para lograr la aceleración dada en b?
  - ¿Cuál es el máximo speedup que se puede lograr resolviendo 5 instancias de este problema con dos procesadores?
- Liste todas las posibles salidas cuando el siguiente código es ejecutado. Suponga que cada asignación es ejecutada en forma atómica. Explique las dependencias encontradas en el programa.

```
j = 0;
k = 0;
forall (i = 1; i <= 2; i++) {           // ejecuta en paralelo
    j = j + 10;
    k = k + 100;
}
printf("i = %d, j = %d, k = %d\n", i, j, k);
```

- Se desea evaluar el siguiente polinomio para todo  $x \in [v, w]$  con  $x \in \mathbb{N}$ :

$$p(x) = a_0x_0 + a_1x_1 + \dots + a_{n-1}x_{n-1}$$

Escriba un pseudocódigo paralelo utilizando la estrategia de descomposición de dominio y otro utilizando descomposición funcional para evaluar el polinomio  $p$  donde el grado  $n$ , los coeficientes  $v$  y  $w$  son la entrada del programa. Indique la complejidad de su programa tomando en cuenta el número de procesadores ( $P$ ), y el tamaño del problema ( $n$ ).

- Sea un conjunto de lectura de instrucciones  $L(S_k) = \{a_1, a_2, \dots, a_n\}$  y un conjunto de escritura  $E(S_k) = \{b_1, b_2, \dots, b_n\}$ .  
Las condiciones de Bernstein definen que dos conjuntos de instrucciones  $S_i$  y  $S_j$  se puede ejecutar concurrentemente si:

- $L(S_i) \cap E(S_j) = \emptyset$
- $E(S_i) \cap L(S_j) = \emptyset$
- $E(S_i) \cap E(S_j) = \emptyset$

- Sea un programa secuencial que consiste de las siguientes cinco instrucciones  $S_1$  hasta  $S_5$ . Considere cada instrucción como un proceso separado, identifique de manera clara el conjunto de entrada y el conjunto de salida para cada proceso.

```

S1: A = B + C
S2: C = B * D
S3: S = 0
S4: DO I = A, 100
      S = S + X(I)
      ENDDO
S5: IF (S >= 1000) C = C * 2

```

Reestructure el programa usando las condiciones de Bernstein para lograr el máximo paralelismo entre dos procesos. Si algún par de procesos no se pueden ejecutar de manera concurrente, especifique cuál(es) de las condiciones de Bernstein son violadas.

Realice un diagrama del grafo de precedencia y reescribalo, si es posible, con las estructuras `fork` y `join`.

- Usando las condiciones de Bernstein, construya un grafo de precedencia del siguiente código y el programa concurrente correspondiente usando el par *cobegin-coend* (para ejecución concurrente de bloques de código)

```
S1: A = B * B
```

S2:  $C = E * A$   
 S3:  $D = F * B$   
 S4:  $Z = C + D$   
 S5:  $Y = Z + W$

- Considere un sistema paralelo homogéneo con 4 equipos con las mismas prestaciones. Se desea evaluar la utilización del algoritmo paralelo de Strassen para el cálculo de la multiplicación de matrices de tamaño  $N \times N$ . Asumiendo que la suma de dos números tiene un costo de una (1) unidad de tiempo, la multiplicación tiene un costo de dos (2) unidades de tiempo, la transferencia por la red de un número tiene un costo de dieciséis (16) unidades de tiempo y que los demás tiempos son despreciables, calcule el número  $N$  mínimo para el cual el sistema paralelo es ventajoso frente al secuencial. A su vez, calcule las medidas de performance vistas en el curso para  $N$  igual a 1, 2, 4, 8, 16 y 32. Realice un análisis a través de las medidas de performance vistas en el curso. Muestre los resultados en tablas y en formato gráfico y justifique los resultados obtenidos.
- Un procesador vectorial con pipeline extiende a la CPU tradicional con registros que pueden almacenar un vector de números de punto flotante e instrucciones que manejan vectores. Por ejemplo, un doble ciclo

```
for i := 1 to 64 do
  for j := 1 to 64 do
    Y[k,j] := a * X[i,j] + Y[k,j];
```

podría implementarse mediante el código

```
R1 := 0
R2 := 0
R3 := k*N           ; N es el largo de fila

for i := 1 to 64 do
  LV  V2,X(R1)      ; cargar fila de X (R1 es i*N)
  MULTSV V3,a,V2    ; multiplicar cada elemento por a
  LV  V1,Y(R2)      ; cargar la k-ésima fila de Y (R2 es k*N)
  ADDV V4,V3,V1     ; suma Y[i] y a*X[i]
  SV  V4,Y(R3)      ; guarda el vector en el array Y

  R1 := R1+N
  R2 := R2+N
end do
```

La arquitectura vectorial clásica surgió con la Cray 1 (1976), y existieron muchos diseños posteriores exitosos por parte de Cray y otros, incluyendo el NEC SX-6 usado en el Earth Simulator, la máquina paralela más potente del mundo en 2005. Las principales ventajas del procesamiento vectorial de registros e instrucciones se relacionan con la eficiencia del procesamiento de pipeline y la capacidad de explotar el acceso a memoria.



Figura 1: El supercomputador Cray-1 (Laboratorio Nacional de Los Álamos, USA).

Suponga que se dispone de una computadora vectorial que permite acelerar los cálculos unas 20 veces (en promedio) con respecto a una ejecución normal (secuencial) cuando se utiliza el modo vectorial.

- ¿Qué porcentaje de código “vectorizable” se necesita para alcanzar un speedup de 2 en la ejecución de una aplicación paralela?
- Estudios estadísticos han mostrado que el porcentaje de código vectorizable es de 70% para un determinado tipo de programas relevantes en las investigaciones en estudio. Se plantea la posibilidad de modificar el hardware vectorial para duplicar su desempeño efectivo (mejorar en un factor de 40 el tiempo de ejecución de código paralelizable), pero con un costo de inversión elevado. Como alternativa, se sugiere invertir un valor razonable en adquirir un compilador que podría mejorar levemente el porcentaje de vectorización de aplicaciones típicas. ¿Qué incremento en el porcentaje de vectorización sería necesario para obtener la misma mejora de desempeño que si se adquiriera la nueva arquitectura de hardware? ¿Qué inversión sugeriría como más razonable en términos de costo y desempeño?
- Grafique los valores máximos de speedup en función del porcentaje de código vectorizable y comente los resultados.