

Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática
Docente: Fred Torres Cruz
Autor: Cliver Wimar Vilca Tinta

Trabajo Encargado - N° 005

Ejercicios Modelo LogP

Ejercicio 1

Supongamos que tenemos un sistema con los siguientes parámetros:

- L: Latencia de la red = 150 microsegundos
- o: Sobrecarga de la comunicación = 25 microsegundos
- g: Gap de comunicación = 30 microsegundos
- P: Número de procesadores = 8

Queremos calcular el tiempo total de comunicación T para enviar $N = 32$ mensajes. Utilizando la fórmula del modelo LogP:

$$T = L + 2 \cdot o + \left(\left\lceil \frac{N}{P} \right\rceil - 1 \right) \cdot g$$

$$\begin{aligned} T &= 150 + 2(25) + \left(\left\lceil \frac{32}{8} \right\rceil - 1 \right) 30 \\ &= 150 + 50 + (4 - 1)30 \\ &= 150 + 50 + 90 \\ &= 290 \text{ microsegundos} \end{aligned}$$

Salida Python:

```
Ingrese los parámetros del modelo LogP:
Latencia (L) en microsegundos: 150
Overhead (o) en microsegundos: 25
Gap (g) en microsegundos: 30
Número de mensajes (N): 32
Número de procesadores (P): 8

Tiempo total de comunicación T: 290 microsegundos
```

Ejercicio 2

Considere un sistema con los siguientes parámetros del modelo LogP:

- L : Latencia de la red = 80 microsegundos
- o : Sobrecarga de la comunicación = 20 microsegundos
- g : Gap de comunicación = 40 microsegundos
- P : Número de procesadores = 16

Calcule el tiempo de ejecución T para una operación de suma de vectores de longitud $n = 10^8$. Según el modelo LogP, el tiempo de ejecución T para la suma de vectores está dado por:

$$T = L + 2 \cdot o + \left(\left\lceil \frac{N}{P} \right\rceil - 1 \right) \cdot g$$

$$\begin{aligned} T &= 80 + 2(20) + \left(\left\lceil \frac{10^8}{16} \right\rceil - 1 \right) 40 \\ &= 80 + 40 + 6249999 \cdot 40 \\ &= 250000080 \text{ microsegundos} \end{aligned}$$

Salida Python:

```
Ingrese los parámetros del modelo LogP:
Latencia (L) en microsegundos: 80
Overhead (o) en microsegundos: 20
Gap (g) en microsegundos: 40
Número de mensajes (N): 100000000
Número de procesadores (P): 16

Tiempo total de comunicación T: 250000080 microsegundos
```

Ejercicio 3

Suponga que se tiene un sistema con los siguientes parámetros del modelo LogP:

- L : Latencia de la red = 120 microsegundos
- o : Sobrecarga de la comunicación = 15 microsegundos
- g : Gap de comunicación = 50 microsegundos
- P : Número de procesadores = 32

Determine el tiempo de ejecución T para una operación de multiplicación de matrices de tamaño $n \times n$, donde $n = 1500$.

Para la multiplicación de matrices de tamaño $n \times n$, el tiempo de ejecución T según el modelo LogP está dado por:

$$T = L + 2 \cdot o + \left(\left\lceil \frac{N}{P} \right\rceil - 1 \right) \cdot g$$

Sustituyendo los valores dados:

$$\begin{aligned} T &= 120 + 2(15) + \left(\left\lceil \frac{1500^2}{32} \right\rceil - 1 \right) 50 \\ &= 120 + 30 + 70312 \cdot 50 \\ &= 120 + 30 + 3515600 \\ &= 3515750 \text{ microsegundos} \end{aligned}$$

Salida Python:

```
Ingrese los parámetros del modelo LogP:
Latencia (L) en microsegundos: 120
Overhead (o) en microsegundos: 15
Gap (g) en microsegundos: 50
Número de mensajes (N): 2250000
Número de procesadores (P): 32

Tiempo total de comunicación T: 3515750 microsegundos
```

Ejercicio 4

Considera una máquina paralela con los siguientes parámetros del modelo LogP:

- L : Latencia de la red = 200 microsegundos
- o : Sobrecarga de la comunicación = 40 microsegundos
- g : Gap de comunicación = 60 microsegundos
- P : Número de procesadores = 64

Calcula el tiempo de ejecución T para una operación de ordenamiento por mezcla (merge sort) sobre un arreglo de longitud $N = 10^9$.

Para el ordenamiento por mezcla de un arreglo de longitud N , el tiempo de ejecución T según el modelo LogP está dado por:

$$T = L + 2 \cdot o + \left(\left\lceil \frac{N}{P} \right\rceil - 1 \right) \cdot g$$

Sustituyendo los valores dados:

$$\begin{aligned}
 T &= 200 + 2(40) + \left(\left\lceil \frac{10^9}{64} \right\rceil - 1 \right) 60 \\
 &= 200 + 80 + 15624999 \cdot 60 \\
 &= 200 + 80 + 937499940 \\
 &= 937500220 \text{ microsegundos}
 \end{aligned}$$

Salida Python:

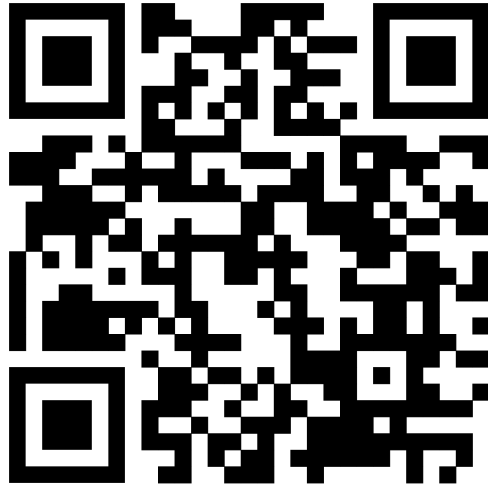
```
Ingrese los parámetros del modelo LogP:
Latencia (L) en microsegundos: 200
Overhead (o) en microsegundos: 40
Gap (g) en microsegundos: 60
Número de mensajes (N): 1000000000
Número de procesadores (P): 64

Tiempo total de comunicación T: 937500220 microsegundos
```

Codigo Python Modelo LogP

```
import math
def logp_model(L, o, g, N, P):
    Calcula el tiempo total de comunicación T según el modelo LogP.
    Parámetros:
    L (int): Latencia
    o (int): Overhead
    g (int): Gap
    N (int): Número de mensajes
    P (int): Número de procesadores
    Retorna:
    T (int): Tiempo total de comunicación en microsegundos
    T = L + 2 * o + (math.ceil(N / P) - 1) * g
    return T
def main():
    print("Ingrese los parámetros del modelo LogP:")
    try:
        L = int(input("Latencia (L) en microsegundos: "))
        o = int(input("Overhead (o) en microsegundos: "))
        g = int(input("Gap (g) en microsegundos: "))
        N = int(input("Número de mensajes (N): "))
        P = int(input("Número de procesadores (P): "))
        T = logp_model(L, o, g, N, P)
        print(f"\nTiempo total de comunicación T: {T} microsegundos")
    except ValueError:
        print("Por favor, ingrese valores enteros válidos.")
if __name__ == "__main__":
    main()
```

Codigo QR y link de Repositrio GitHub



Link Repositorio GutHub https://github.com/CliverVilca/Parallel-Computing/blob/main/Modelo_Logica