Desarrollo de Software en Arquitecturas Paralelas

Práctica 2: TCOM

Elza Sarrías Alieva

Contenido

1. Introducción

El objetivo de este trabajo consiste en evaluar el coste de envío de mensajes en el laboratorio de la universidad. Para ello, se van a estimar los valores de los parámetros $\boldsymbol{\beta}$ y $\boldsymbol{\tau}$, que indican respectivamente la latencia necesaria para el envío de un mensaje y el tiempo para enviar un byte de datos.

De esta manera, el coste de comunicaciones entre dos procesadores equivale al tiempo que se tarda para establecer la conexión más el tiempo necesario para transmitir el mensaje completo. Se puede expresar de la siguiente manera:

$$T_{com} = \beta + \tau \cdot NBytes$$

2. Estimación de $\boldsymbol{\beta}$

Para estimar el tiempo necesario para establecer la conexión se debería enviar un mensaje vacío. Sin embargo, dado que esto no es posible en MPI, se utilizarán mensajes con el tamaño mínimo permitido: un byte.

Dado que sería imposible sincronizar perfectamente los relojes de los procesadores con los que se va a trabajar, para la medición de tiempo se van a enviar dos mensajes: uno de ida y otra de vuelta; y dividir el tiempo necesario para ambos envíos entre dos. Cada par de envíos se realizará 1000 repeticiones y la aproximación final será la media aritmética entre ellas.

$$\beta \approx \frac{\sum_{i=1}^{1000} \left(\frac{T_{ida_i} + T_{vuelta_i}}{2} \right)}{1000}$$

De esta manera, el procedimiento consistirá en que el proceso 0 envie un mensaje compuesto por un byte al proceso 1 y se quede a la espera de recibir el mensaje de vuelta. El proceso 1, tras recibir el mensaje inicial enviará un mensaje de vuelta del mismo tamaño. Esto será repetido 1000 veces y el proceso 0 irá acumulado los tiempo necesarios para realizar los envíos, de forma que una vez finalice el proceso se calcule la media aritmética de los tiempos.

3. Estimación de $\boldsymbol{\tau}$

A la hora de estimar el valor de $\boldsymbol{\tau}$, no basta con hacer una única medición, ya que el protocolo TCP/IP fragmenta los mensajes en función de su tamaño, por lo que el tiempo necesario para enviar un byte no será siempre el mismo. Por tanto, se realizan mediciones de $\boldsymbol{\tau}$ con diferentes tamaños de mensajes: 256 B, 512 B, 1 KB, 2 KB, 4 KB, 8 KB, 16 KB, 32 KB, 64 KB, 128 KB, 256 KB, 215 KB, 1 MB, 2 MB, 4 MB.

Cabe destacar que para el envío de mensajes en MPI se utilizada el tipo MPI_DOUBLE, que corresponde a 8 bytes. Por tanto, para un mensaje de tamaño N, se enviarán N/8 doubles. De esta forma, los mensajes a enviar tendrán tamaños entre 2^5 doubles (256B) y 2^{19} doubles (4MB).

El procedimiento para medir el tiempo de comunicación es el mismo que para la estimación de $\boldsymbol{\beta}$: mensajes de ida y vuelta, con 1000 repeticiones para cada mensaje. La diferencia es que ahora este procedimiento se realizará para tamaños diferentes de mensajes. De esta manera, la estimación de $\boldsymbol{\tau}$ para un tamaño de mensaje determinado viene dado por la siguiente expresión:

$$\tau \approx \frac{T_{com} - \beta}{NBytes}$$

$$T_{com} = \frac{\sum_{i=1}^{1000} \left(\frac{T_{ida_i} + T_{vuelta_i}}{2} \right)}{1000}$$

4. Implementación

La implementación del programa que realiza las mediciones explicadas anteriormente se puede resumir en el siguiente pseudocódigo:

```
Inicialización variables y entorno ejecución de MPI

if (proceso==0)

  // Estimación de Beta
  for i=0..1000
    startTime=MPI_Wtime() //Inicio
    MPI_Send(tipo=MPI_Byte, tamaño=1, destino=1) // Envío
    MPI_Recv(tipo=MPI_Byte, tamaño=1, origen=1) // Respuesta
    endTime=MPI_Wtime() // Final
    totalTime+=(endTime-startTime)/2 // Tiempo acumulado
  beta=totalTime/1000 // Beta = media aritmética tiempos

  // Estimación de Tau
  for i=5..19 // Tamaño de mensajes de 2^5 a 2^19
    ndoubles=pow(2, i) // Número de doubles a enviar y recibir
    nbytes=ndoubles*8 // Número de bytes correspondiente
    totalTime=0
    for j=0..1000
      startTime=MPI_Wtime() // Inicio
      MPI_Send(tipo=MPI_Double, tamaño=ndoubles, destino=1) // Envío
      MPI_Recv(tipo=MPI_Double, tamaño=ndoubles, origen=1) // Respuesta
      endTime=MPI_Wtime() // Final
      totalTime+=(endTime-startTime)/2 // Tiempo acumulado
    tcom[i]=totalTime/1000 // Tcom del mensaje i = media aritmética tiempos
    tau[i]=(tcom[i]-beta)/nbytes // Tau i = (Tcom - Beta)/NBytes

else if (proceso==1)

  // Estimación de Beta
  for i=0..1000
    MPI_Recv(tipo=MPI_Byte, tamaño=1, origen=0) // Respuesta
    MPI_Send(tipo=MPI_Byte, tamaño=1, destino=0) // Envío

  // Estimación de Tau
  for i=5..19 // Tamaño de mensajes de 2^5 a 2^19
    ndoubles=pow(2, i) // Número de doubles a recibir y enviar
    for j=0..1000
      MPI_Recv(tipo=MPI_Double, tamaño=ndoubles, origen=0) // Respuesta
      MPI_Send(tipo=MPI_Double, tamaño=ndoubles, destino=0) // Envío
```

5. Resultados

5.1. Entorno distribuido

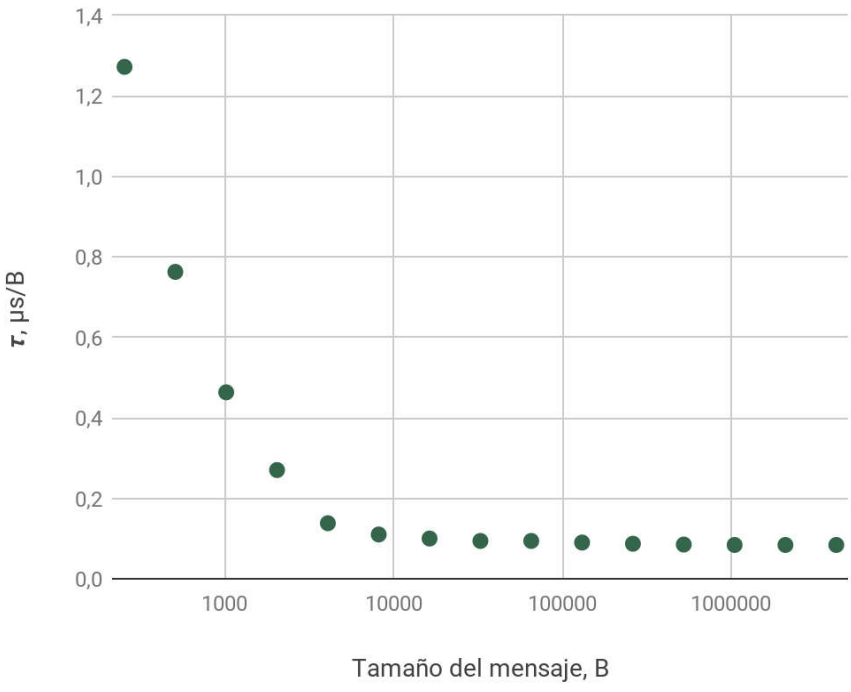
Para evaluar el coste de comunicaciones en el laboratorio se ha hecho uso de dos nodos situados en computadoras diferentes. Para ello, se debe crear un archivo que especifique los nombres de las computadoras que ejecutarán el código, e indicar que cada computadora sólo debe utilizar un núcleo. Para empezar la ejecución se puede utilizar el atajo `make run_remote` que ejecuta el siguiente comando:

```
mpirun -machinefile machinefile -np 2 -npernode 1 tcom
```

A continuación, se pueden ver los resultados de la prueba en el entorno distribuido. Se puede observar que con los mensajes más grandes el valor de τ es más pequeño. Es decir, el rendimiento es mayor ya que el tiempo necesario para transmitir un byte de datos disminuye conforme aumenta el tamaño del mensaje. Se debe a que con el protocolo TCP/IP el aprovechamiento de la red es mayor con mensajes más grandes.

Entorno distribuido

β		35,483 µs
τ	256 B	1,272 µs/B
	512 B	0,763 µs/B
	1 KB	0,464 µs/B
	2 KB	0,271 µs/B
	4 KB	0,139 µs/B
	8 KB	0,111 µs/B
	16 KB	0,101 µs/B
	32 KB	0,095 µs/B
	64 KB	0,095 µs/B
	128 KB	0,091 µs/B
	256 KB	0,088 µs/B
	512 KB	0,086 µs/B
	1 MB	0,085 µs/B
	2 MB	0,085 µs/B
	4 MB	0,085 µs/B

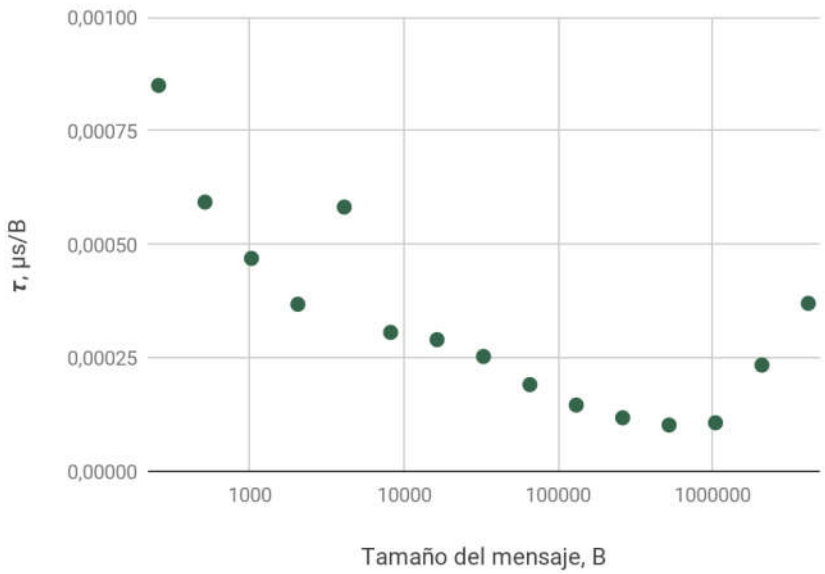


5.2. Entorno local

Por otro lado, también se ha realizado evaluación de coste de comunicación en un entorno local. Es decir, en este caso los nodos del programa son dos núcleos de la misma computadora. Para ejecutar esta prueba se puede utilizar el atajo `make run`, que ejecuta el siguiente comando:

```
mpirun -np 2 tcom
```

En la siguiente tabla se pueden ver los resultados de la prueba en el entorno local. Se puede observar que el tiempo necesario para la transmisión de datos es considerablemente menor que en el entorno distribuido, ya que en este caso la información no se envía por la red de comunicación. Además, la diferencia entre los valores de τ para diferentes tamaños de mensajes es muy pequeña, ya que la latencia para escribir en la memoria principal es insignificante.

Entorno local			
β		0,422 µs	
τ	256 B	0,000850 µs/B	
	512 B	0,000593 µs/B	
	1 KB	0,000469 µs/B	
	2 KB	0,000368 µs/B	
	4 KB	0,000582 µs/B	
	8 KB	0,000306 µs/B	
	16 KB	0,000290 µs/B	
	32 KB	0,000253 µs/B	
	64 KB	0,000191 µs/B	
	128 KB	0,000146 µs/B	
	256 KB	0,000118 µs/B	
	512 KB	0,000102 µs/B	
	1 MB	0,000107 µs/B	
	2 MB	0,000234 µs/B	
	4 MB	0,000370 µs/B	