Práctica 2.A: Tiempo de comunicación

Juan José Conejero
jjcs2@alu.ua.es

Desarrollo Software en Arquitecturas Paralelas.

Resumen Cuando se establece una comunicación mediante protocolo MPI, comúnmente, no se tiene en cuenta el coste de las comunicaciones. Realmente, no es necesario si la latencia de la red es baja o no infiere mucho a la hora de establecer resultados de un experimento de dado. No obstante, pueden darse experimentos en los que se necesite restar la latencia de comunicación para determinar el determinado coste de un algoritmo, sin comunicación. En esta práctica se aborda el problema del cálculo del coste de comunicación.

1. Introducción

El objetivo de este ejercicio es evaluar los valores de los parametros que determinan el modelo de coste de las comunicaciones en la plataforma del laboratorio. Se verá el coste de las comunicaciones entre dos procesadores determinados. Los parametros β y τ indican respectivamente la latencia necesaria para el envío de un mensaje y el tiempo necesario para enviar un byte. En la **Figura 1** se puede ver los parámetros y su relación que conforman este coste de comunicación. Además, estos parametros se pueden estimar de la siguiente forma:

- El valor de β sera el tiempo necesario en enviar un mensaje sin datos.
- El valor de τ sera el tiempo requerido para enviar un mensaje menos el tiempo de latencia, dividido por el numero de bytes del mensaje.

$$T_{com} = \beta + \tau \cdot Tam\_Mensaje$$

Figura 1: Coste de comunicación entre dos procesos

2. Implementación

Es sabido que los protocolos de comunicación, para el control y la verificación de paquetes, insertan sus propias cabeceras y fragmentan estos paquetes en función del protocolo que sigan. Además, el propio **sistema operativo** sigue

Algoritmo 1 Estimación de la latencia β (Emisor)

```
1  char byteEnviado='0';
2  for (iEnvios=0; iEnvios<enviosTotales; iEnvios++){
3    tiempoIni=MPI_Wtime();
4    MPI_Send(&bytesEnviado, 1, MPI_BYTE, 1, 123,
         MPI_COMM_WORLD);
5    MPI_Recv(&bytesEnviado, 1, MPI_BYTE, 1, 321,
         MPI_COMM_WORLD, &estado);
6    tiempoFin=MPI_Wtime();
7    tiempoIdaVuelta=(tiempoFin-tiempoIni)/2;
8    mediaTiempo+=tiempoIdaVuelta;
9  }
10 beta=mediaTiempo/enviosTotales;
```

una serie de instrucciones antes de enviar los paquetes, que harán que estos se retrasen. Por tanto, en el experimento se realizarán dos pasos: establecer la latencia de la comunicación, y el tiempo que tarda un byte en transmitirse de una máquina a otra.

2.1. Análisis de la latencia

Como se ha mencionado anteriormente, el sistema operativo y la red de comunicación, invierten parte del tiempo en realizar tareas y protocolos de comunicación además de los datos de comunicación en sí. Es por eso, que a la hora de realizar los cálculos de un experimento de comunicación, se debe sustraer este tiempo a cada comunicación.

Para establecer ese tiempo, una posible imeplementación, es la de enviar un paquete sin datos (en nuestro caso, con el mínimo número de datos posible) y tomar ese dato como latencia de comunicación.

En el caso de MPI Send y MPI Recv no admiten un mensaje sin datos, así que la solución para estimar el valor de la latencia es comunicar un solo byte, usando el tipo MPI BYTE, a partir de por ejemplo una variable del tipo **char.** Una posible implementación es la que figuran en los **Algoritmos 1 y 2.** Como se puede observar, este algoritmo se realizar **numVeces** ocasiones y posteriormente se hace una media de ese tiempo de comunicación; esto se hace para estimar una media aproximada y no depender de un solo dato.

2.2. Análisis de la transmisión

Teniendo en cuenta que se utiliza el protocolo **TCP/IP** como medio de transmisión de los mensajes, realmente el valor del tiempo de transferencia será diferente para mensajes pequeños y grandes, ya que el protocolo de comunicaciones fragmenta los mensajes en bloques. Por tanto, se debe establecer una

Algoritmo 2 Estimación de la latencia β (Receptor)

```
1 for (iEnvios=0; iEnvios<enviosTotales; iEnvios++){
2   MPI_Recv(&bytesEnviado, 1, MPI_BYTE, 0, 123,
        MPI_COMM_WORLD, &estado);
3     MPI_Send(&bytesEnviado, 1, MPI_BYTE, 0, 321,
          MPI_COMM_WORLD);
4 }
```

Algoritmo 3 Cálculo τ de a partir de t_com y β (Emisor)

```
1 for(jBit=BITS_INICIALES; jBit<=BITS_FINALES; jBit++){
2   mediaTiempo=0, t_com=0;
3   tam_mensaje=(int)pow(2, jBit);
4   for (iEnvios=0; iEnvios<enviosTotales; iEnvios++){
5     tiempoIni=MPI_Wtime();
6     MPI_Send(&mensajeGrande, tam_mensaje, MPI_DOUBLE, 1,
          123, MPI_COMM_WORLD);
7     MPI_Recv(&mensajeGrande, tam_mensaje, MPI_DOUBLE, 1,
          321, MPI_COMM_WORLD, &estado);
8     tiempoFin=MPI_Wtime();
9     tiempoIdaVuelta=(tiempoFin-tiempoIni)/2;
10    mediaTiempo+=tiempoIdaVuelta;
11  }
12   t_com=mediaTiempo/enviosTotales;
13  tau=t_com-beta;
14  tau=tau/(tam_mensaje*8);
15 }
```

comunicación igual que la anterior (Estimación de la latencia) pero enviando **paquetes de datos** más grandes.

Para estimar un valor adecuado, se irán enviando paquetes de datos que abarcarán desde 2^5 hasta 2^{19} **doubles.** Puesto que cada double son **8 bytes**, por cada envío estaremos enviando $2^x \cdot$ **8 bytes.** Esto, nos da unos análisis en los datos que abarcan desde **256 bytes** hasta **4096 Kbytes.** En los **algoritmos 3 y 4** se puede apreciar como se realiza el proceso de envío de bytes para calcular τ de a partir de t_com y β

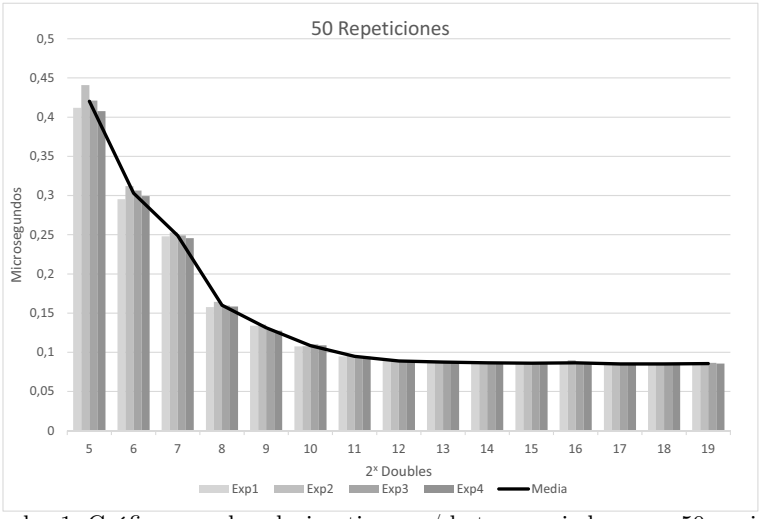
Algoritmo 4 Cálculo τ de a partir de t_com y β (Receptor)

```
1 for(jBit=BITS_INICIALES; jBit<=BITS_FINALES; jBit++){
2   tam_mensaje=(int)pow(2, jBit);
3   for (iEnvios=0; iEnvios<enviosTotales; iEnvios++){
4     MPI_Recv(&mensajeGrande, tam_mensaje, MPI_DOUBLE,0,
          123, MPI_COMM_WORLD, &estado);
5     MPI_Send(&mensajeGrande, tam_mensaje, MPI_DOUBLE, 0,
          321, MPI_COMM_WORLD);
6    }
7 }
```

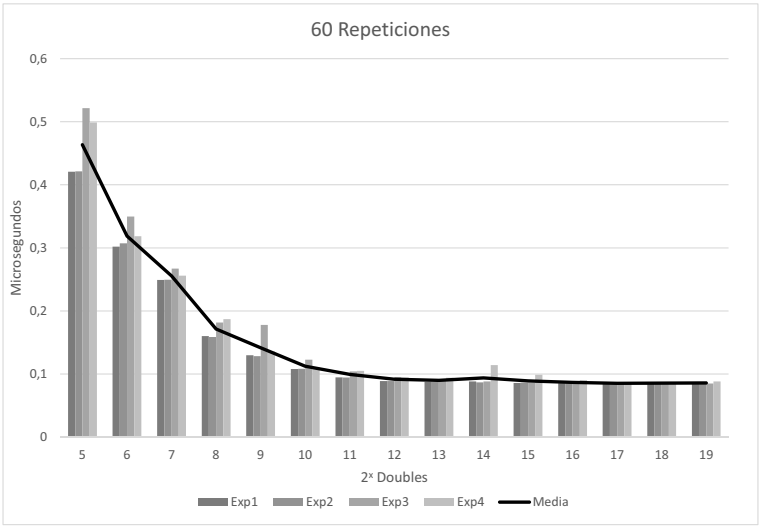
3. Análisis y conclusión

Finalmente, a la vista de los resultados obtenidos, se puede concluir que el tiempo de latencia que existe entre dos máquinas del laboratorio L01 de la EPS de la Universidad de Alicante tiende siempre al mismo valor. En las gráficas que se muestran en los **cuadros 1, 2, 3, 4, 5 y 6** se pueden analizar los datos que surgen de realizar el experimento con 50, 60, 70, 80, 90 y 100 envíos respectivamente.

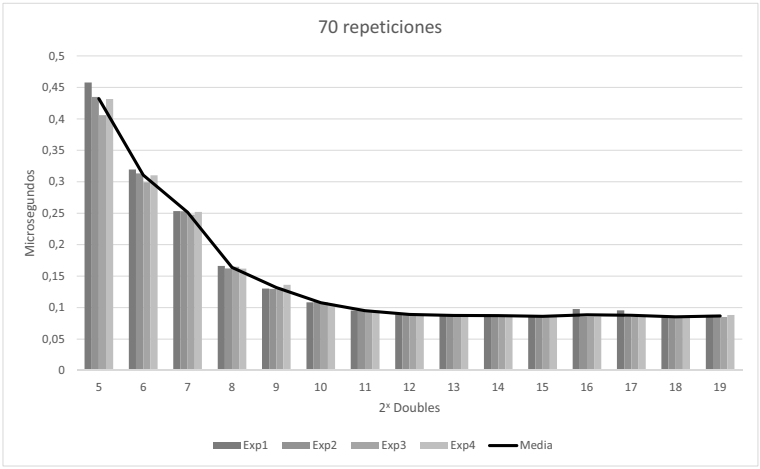
Finalmente, se puede establecer que **el tiempo medio necesario para enviar un byte por la red LAN** es de $0,086ms$. Además, la variación de los valores independientemente del número de envios que hagamos, cuando se realiza la comunicación con pocos bytes, nos da una idea de la gran cantidad de datos de los diversos protocolos que tiene una sola trama.



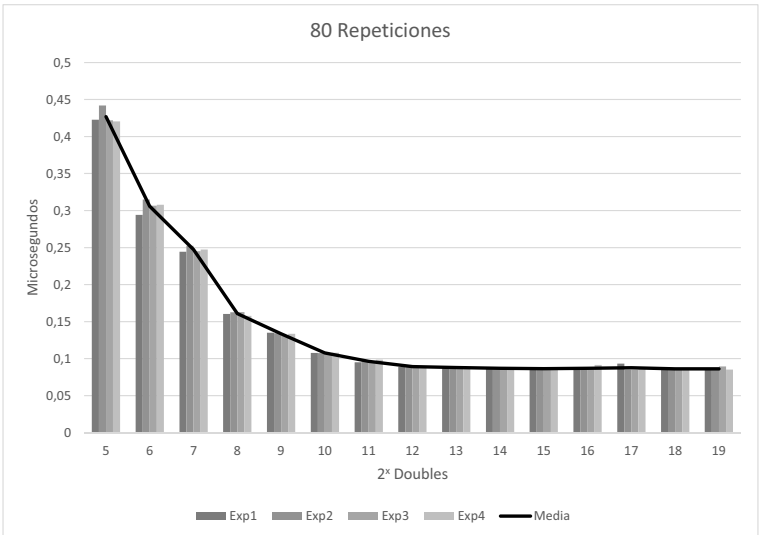
Cuadro 1: Gráficas con la relacion tiempo / bytes_enviados para 50 envios



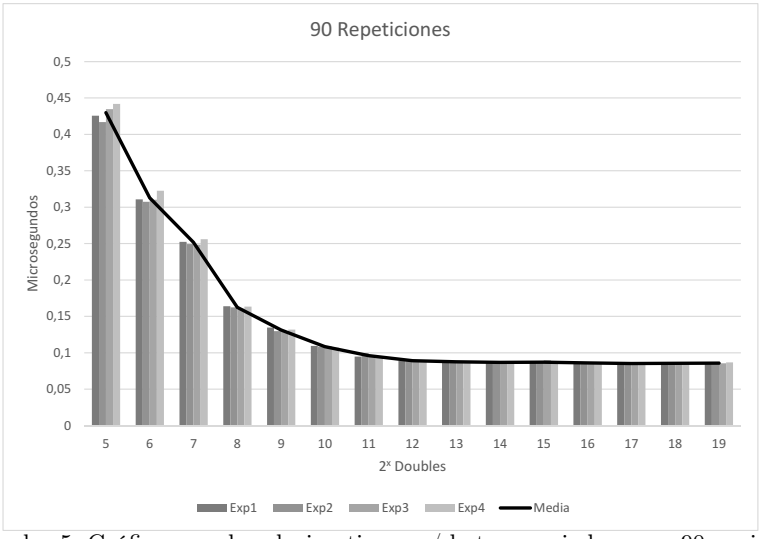
Cuadro 2: Gráficas con la relacion tiempo / bytes_enviados para 60 envios



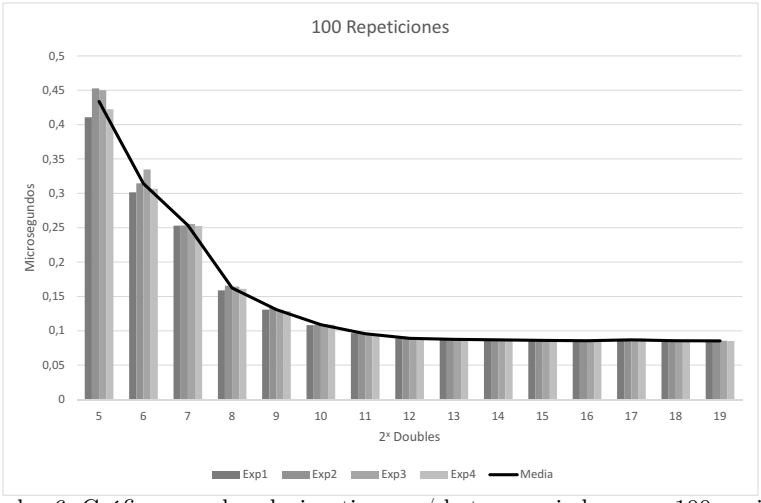
Cuadro 3: Gráficas con la relacion tiempo / bytes_enviados para 70 envios



Cuadro 4: Gráficas con la relacion tiempo / bytes_enviados para 80 envios



Cuadro 5: Gráficas con la relacion tiempo / bytes_enviados para 90 envios



Cuadro 6: Gráficas con la relacion tiempo / bytes_enviados para 100 envios