


MPI PRACTICA 1

1. RED TOROIDE

Dado un archivo con nombre datos.dat, cuyo contenido es una lista de valores separados por comas, nuestro programa realizará lo siguiente:

El proceso de rank 0 distribuirá a cada uno de los nodos de un toroide de lado L, los $L \times L$ números reales que estarán contenidos en el archivo datos.dat. 

En caso de que no se hayan lanzado suficientes elementos de proceso para los datos del programa, éste emitirá un error y todos los procesos finalizarán.

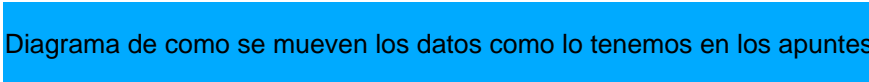
En caso de que todos los procesos han recibido su correspondiente elemento, comenzará el proceso normal del programa.

Se pide calcular el elemento menor de toda la red, el elemento de proceso con rank 0 mostrará en su salida estándar el valor obtenido.

La complejidad del algoritmo no superará $O(\text{raiz_cuadrada}(n))$ Con n número de elementos de la red.

Se pide enunciado del problema, documentación e implementación en formato electrónico.

La documentación debe incluir:

- Enunciado del problema
- Planteamiento de solución
- Diseño de programa
- Explicación de flujo de datos en la red para cada comando MPI usado en la solución.  Diagrama de como se mueven los datos como lo tenemos en los apuntes
- Fuentes del programa
- Instrucciones de cómo compilar y ejecutar.
- Conclusiones

El archivo de entrega debe incluir:

- Archivo Makefile
- Fuentes C
- Binarios para plataforma Linux intel-386 – openmpi. Resultado de la compilación
- Documentación en formato electrónico
- Todo incluido en una carpeta cuyo nombre es el del alumno.

2. RED HIPERCUBO

Dado un archivo con nombre datos.dat, cuyo contenido es una lista de valores separados por comas, nuestro programa realizará lo siguiente:

El proceso de rank 0 distribuirá a cada uno de los nodos de un Hiper cubo de dimensión D , los 2^D números reales que estarán contenidos en el archivo datos.dat. En caso de que no se hayan lanzado suficientes elementos de proceso para los datos del programa, éste emitirá un error y todos los procesos finalizarán.

En caso de que todos los procesos han recibido su correspondiente elemento, comenzará el proceso normal del programa.

Se pide calcular el elemento mayor de toda la red, el elemento de proceso con rank 0 mostrará en su salida estándar el valor obtenido. La complejidad del

algoritmo no superará $O(\log_2(n))$ Con n número de elementos de la red.

Se pide enunciado del problema, documentación e implementación tanto en formato electrónico como en papel.

La documentación debe incluir:

- Enunciado del problema
- Planteamiento de solución
- Diseño de programa
- Explicación de flujo de datos en la red para cada comando MPI usado en la solución.

- Fuentes del programa
- Instrucciones de cómo compilar y ejecutar.
- Conclusiones

El archivo de entrega debe incluir:

- Archivo Makefile
- Fuentes C
- Binarios para pl