PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC 3524 — Tópicos Avanzados en Sistemas Distribuidos – 2017/1

Tarea 2 – MPI

Fecha de Entrega: Miércoles 21-Junio, 23:59

Composición de trabajo: individual

En esta tarea deben utilizar la librería MPI para ejecutar un algoritmo del estilo *branch-and-bound* de manera distribuida y efectuar comparaciones con su versión secuencial.

El Problema

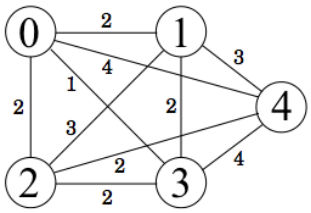
El problema a resolver es el *Wandering Salesman Problem* (WSP). En este problema una persona debe visitar **exactamente una vez todas** las ciudades de un mapa a partir de la posición inicial recorriendo la menor distancia posible. Es muy similar al *Traveling Salesman Problem* (TSP) con la diferencia que el WSP no regresa a la ciudad de origen.

Se trata de un problem NP-Hard, donde, para N ciudades, hay $(N-1)!$ rutas posibles desde una ciudad inicial.

La entrada será una matriz $M_{N\times N} = (d_{i,j}), 0 \leq i, j < N,\ d_{i,j} > 0$, donde $d_{i,j}$ es la distancia entre la ciudad i y la ciudad j. La ciudad inicial siempre será la ciudad 0.

Archivo de entrada

La entrada debe ser recibida por línea de comandos en un archivo de texto. La primera línea de este archivo contendrá un valor $N > 1$ que indica la dimensión de la matriz. A continuación habrá $N-1$ líneas donde cada línea $k, 0 \leq k < N-1$, contendrá $N-1-k$ enteros separados por un espacio indicando el costo entre la ciudad k y la ciudad $p, k < p \leq N-1$. Por ejemplo, para el siguiente mapa:



El archivo de entrada es:

```
5
2 2 1 4
3 2 3
2 2
4
```

Solución *branch-and-bound*

Una solución exhaustiva para el WSP consiste en generar cada una de las combinaciones posibles, calcular su costo, y quedarse con la más corta. Una solución *branch-and-bound* es una solución que inicia un recorrido exhaustivo, pero mantiene una cota (*bound*) para la ruta más corta encontrada hasta el momento y evita explorar rutas cuando el costo parcial de la ruta nueva es menor a esa cota.

En esta tarea debe implementar una solución del tipo *branch-and-bound* para el WSP, utilizando MPI en C.

Infraestructura

La solución debe ser capaz de funcionar en el cluster GRIMA. Para ello debe lanzar su proceso desde el nodo `tripio`, y utilizar como nodos de cómputo a `tripio`, `titan`, `caleuche`, `trauco`, `makemake`, con un máximo de 4 cores en cada uno. No debe ejecutar procesos de cómputo en `hercules` o en otros nodos fuera de los mencionados.

Es recomendable desarrollar su tarea localmente y probar con múltiples procesos locales y solamente ejecutar en el cluster para obtener datos de ejecución. Sea consciente que varios usuarios pueden estar solicitando los mismos recursos y no deje su ejecución para último momento. Puede revisar en todo momento el uso del cluster en esta dirección: `http://hercules.ing.puc.cl/ganglia`

Análisis

Debe efectuar un análisis comparando el tiempo de ejecución, *speedup* y eficiencia para diferentes valores de N y cantidad de procesos MPI, y reportar gráficamente estos valores. Utilice valores de N representativos que permitan apreciar una diferencia en tiempo de ejecución.

Para efectos de comparación, la versión *baseline* debe ser la versión MPI ejecutado usando solamente un proceso local.

Debe escribir un informe donde describa sus experimentos y discuta los resultados.

Restricciones

- El programa debe funcionar usando C en MPI en los nodos del cluster GRIMA. No sirve que funcione únicamente en su computador local.
- El reporte debe entregarse como un archivo `.pdf`.
- Debe incluir un archivo `readme` con instrucciones de ejecución, y un Makefile para efectuar la compilación.
- La entrega se efectuará mediante una carpeta que se llama **exactamente** `T2` en su directorio `home`.
- **NO** entregar código compilado.

Evaluación

- 10 %. Restricciones de entrega (si no se cumplen suficiente condiciones, no se revisa el resto).
- 10 %. Funcionamiento de versión *baseline* (secuencial).
- 25 %. Funcionamiento de versión paralela en un nodo
- 25 %. Funcionamiento de versión paralela en múltiples nodos
- 30 %. Análisis de rendimiento.