Programación Concurrente

Algoritmos Distribuidos

Alfredo Ucendo Arroyo

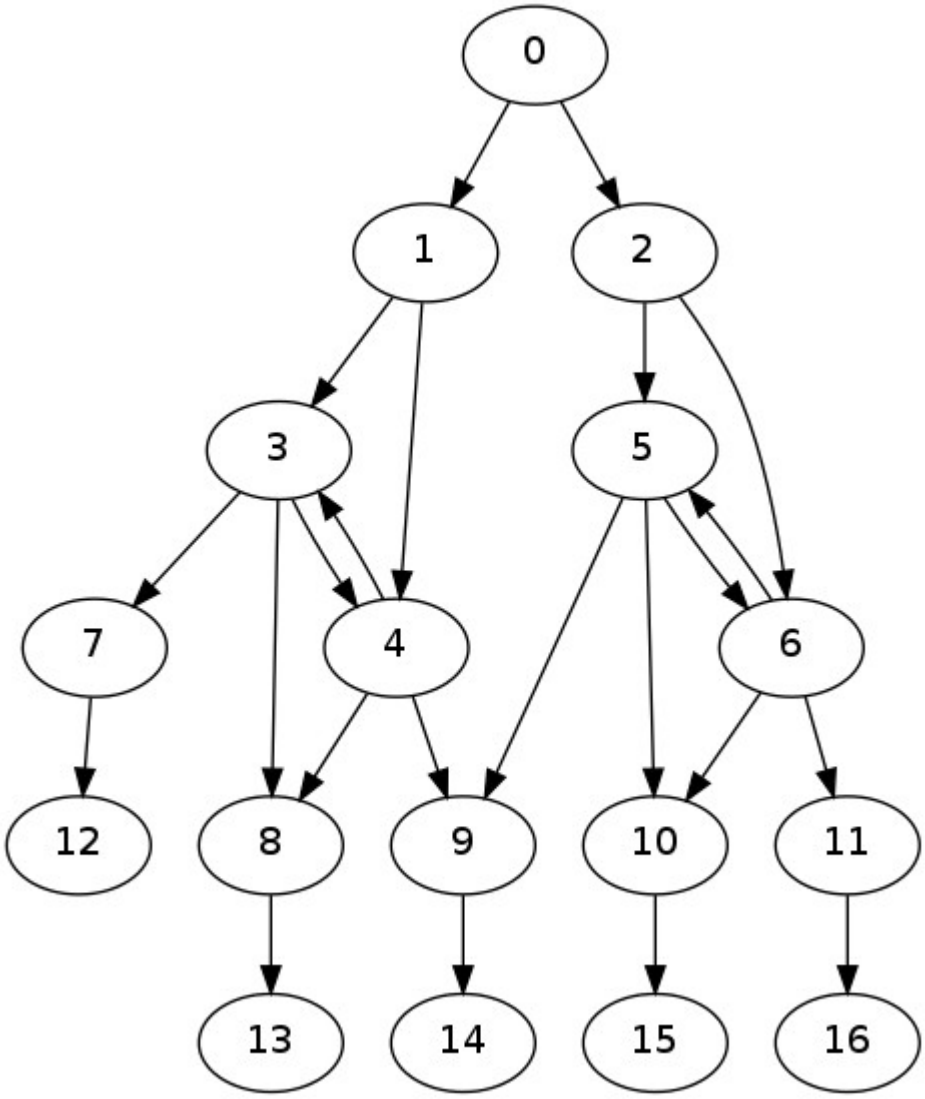
&

Pablo Riutort Grande

Introducción

El objetivo es programar y probar el funcionamiento del algoritmo de terminación distribuida de Dijkstra-Scholten . Para hacerlo, se debe programar mediante hilos y la comunicación entre estos deberá ser con cola del mensaje que proporciona el beanstalkd.

La red de nodos vendrá dada por un grafo que el programa deberá leer e interpretar sus nodos y conexiones:



Llamaremos al nodo raíz nodo *entorno*. Este nodo deberá enviar a sus sucesores una tarea a realizar mediante mensajes. Los nodos sucesores, a su vez deberán repartir el trabajo que les ha enviado su predecesor en el grafo a sus nodos adyacentes de forma que el trabajo se repartirá entre todo el árbol, a estos nodos los incluiremos en *no entorno.*
Una vez que un nodo haya terminado el trabajo, le enviará un signal a su nodo predecesor (padre) indicándole que ha terminado. Un nodo que reciba su primer mensaje de otro nodo, considerará a este su padre. Un nodo no ha terminado de trabajar hasta que sus hijos lo hayan hecho.

Ejecución

La ejecución del programa consiste en enviar un trabajo en una decena de iteraciones.
Se pide:

- Número de mensajes generados en total
- Tiempo que se ha tardado en cada iteración
- Árbol de expansión mínima en cada iteración

Resultados

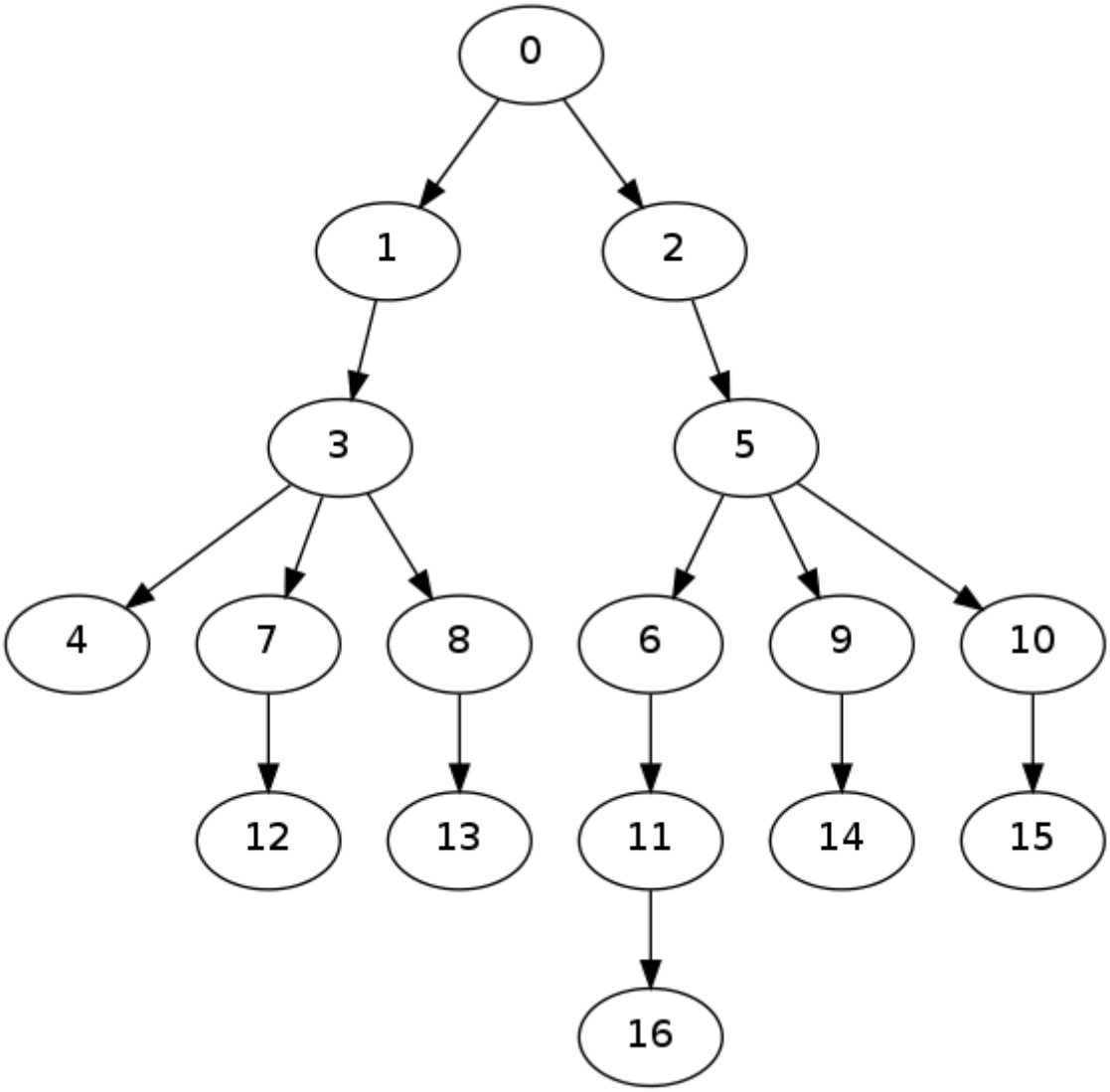
Se han ejecutado 10 iteraciones con los siguientes resultados:

	Tiempo (en ms)	**Mensajes enviados**
Iteración 1	170	123
Iteración 2	21	123
Iteración 3	8	123
Iteración 4	16	123
Iteración 5	7	123
Iteración 6	6	123
Iteración 7	21	123
Iteración 8	30	123
Iteración 9	58	123
Iteración 10	30	123

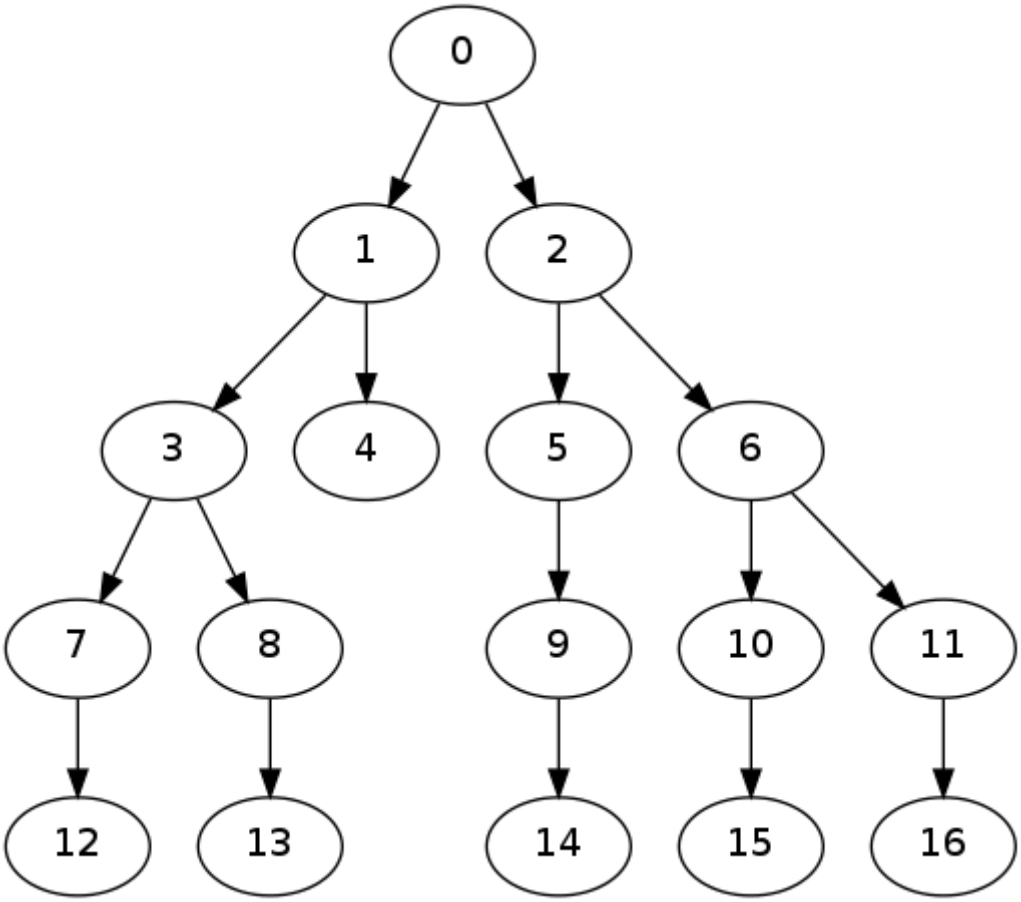
Spanning Tree

La ejecución de las 10 iteraciones han resultado en los siguientes árboles:

Árbol de expansión mínima 1



Árbol de expansión mínima 2



Ambos árboles se han generado en ejecuciones diferentes de la práctica, es decir, unas 10 iteraciones han dado un árbol de expansión mínima y haciendo una segunda ejecución de la práctica (otras 10 iteraciones) han dado otro árbol, pero siempre el mismo en todas las iteraciones. No sabemos el porqué de este hecho, pero somos conscientes que en 10 iteraciones, una de ellas podría dar un árbol de expansión mínima diferente.

Desarrollo de la práctica

Para la ejeción de la práctica hemos optado por leer una vez el archivo adjunto `graph.dot` donde viene el grafo a seguir en el algoritmo. Una vez leído el grafo seremos capaces de crear los links que hay entre cada nodo y saber a qué nodo está conectado cada uno y efectuar los trabajos de manera adecuada.
Hemos tenido algunos problemas con el beanstalkd. Para ejecutar correctamente la práctica hay que ejecutar el comando:

```
beanstalkd -l localhost -p 11300
```

Seguidamente ejecutar el código **dos veces** (la primera siempre da error). Creemos que el responsable de que la primera ejecución siempre falle debe ser el beanstalkd.

La ejecución de la práctica genera los siguientes archivos y ficheros:

1. `resultados.txt`; En este fichero se encuentran los resultados de las 10 iteraciones con información de cada una. En el podemos encontrar cuantos mensajes se han enviado durante cada iteración y lo que ha tardado.. También aparece la estructura del árbol de expansión mínima (como ya se ha comentado antes, siempre el mismo en todas las iteraciones).
2. `spanningTrees/`; En este directorio encontramos la colección de árboles de expansión mínima de cada iteración en formato .dot
3. `spanningTreeX.dot`; Un .dot con el árbol de expansión mínima generado en cada iteración, habrá 10 en total. La X indica en qué iteración ha sido generado.

Otros documentos adjuntos

En esta práctica se adjunta `documento.pdf` y un script llamado `dotToPng` (necesita permisos de ejecución) que pasa a formato .png los árboles de expansión mínima contenidos en el directorio de spanningTrees. También se adjunta `README.md`; Archivo que resume a grandes rasgos el contenido de los fuentes que conforman el proyecto

También disponible en: https://github.com/grimheader/AlgoritmosDistribuidos.git