ST0237 - Proyecto Especial de 3 Créditos
Trabajo Final

Juan G. Lalinde-Pulido

2016-2

1. Introducción

El objetivo final de la materia *ST0237 Proyecto Especial de 3 créditos* es que construyan un cluster simple y desarrollen una aplicación que aproveche el cluster para llevar a cabo las operaciones matemáticas. En este trabajo deben implementar una aplicación utilizando MPI, la cual debe ser ejecutada utilizando el sistema de colas del cluster.

2. Enunciado

El PageRank es un algoritmo para clasificar, i.e. establecer un *ranking*, las páginas web que se ajustaban a una consulta. Fue desarrollado por Larry Page y Segey Brin y está protegido por la patente US 6285999 B1[2]. La descripción detallada de su funcionamiento la pueden encontrar en el libro [1]. El PageRank se puede interpretar de muchas maneras diferentes, pero tal vez la más intuitiva es que el PageRank asociado a un nodo es la probabilidad de que se llegue a él al recorre aleatoriamente la web. La red se representa mediante una matriz de adyancencia A, en la cual la entrada $A[i,j] \neq 0$ si hay un enlace que apunta de la página i a la página j. En el modelo de PageRank de Google, si la página i tiene m enlaces, entonces el valor de cada entrada $A[i,j]$ diferente de cero es $1/m$. Si N es el número de nodos en la red, la matriz A debe ser una matriz estocástica de $N \times N$. El hecho de que la matriz sea estocastica quiere decir que la suma de los elementos de cualquier columna debe dar uno. Es decir:

$$\forall k, 0 \leq k < N, \sum_{i=0}^{N} a_{i,k} = 1 \tag{1}$$

Es evidente que el modelo de Google donde $A[i,j] = 1/m$ si hay un enlace de i a j y el número total de enlaces que salen de i es m, garantiza que se cumple la propiedad (1).

Por otra parte, el vector r, que es de tamaño $N \times 1$, tiene el valor de PageRank para cada nodo y es cumple que

$$Ar = r \tag{2}$$

Si bien hay muchas maneras de calcular el PageRank, la más eficiente utiliza el hecho de que r es un punto fijo de Ar. Para su cálculo se utiliza la siguiente recurrencia:

$$r_{n+1} = Ar_n \tag{3}$$

La condición inicial r_0 puede ser cualquiera, pero normalmente se asume que el todos los elementos del vector r_0 son iguales y tienen el valor $1/N$. Si la notación $r[i]$ expresa el i-ésimo elemento de r, entonces la condición inicial se puede expresar como

$$\forall i, 0 \leq i < N, r_0[i] = 1/N \tag{4}$$

La idea, entonces, es calcular la serie $r_0, r_1, \ldots, r_k$. El criterio de parada es que la diferencia entre r_{k-1} y r_k es menor que una tolerancia dada. De manera más precisa, si ϵ es la tolerancia, entonces se suspende el cálculo de la recurrencia cuando

$$\epsilon < \max_{0 \leq i < N} (|(r_k[i] - r_{k-1}[i]|) \tag{5}$$

Su tarea es implementar el algoritmo de PageRank utilizando MPI, de manera que aproveche eficientemente todos los nodos del cluster para llevar a cabo los cálculos. El programa debe recibir como parámetros el número de nodos N que tiene la red, la densidad d y la tolerancia ϵ. Debe generar una matriz estocástica A de $N \times N$ que tenga una densidad d. Si n es el número de entradas en la matriz diferentes de cero, entonces debe cumplirse que

$$\frac{n}{N^2} \approx d \tag{6}$$

Una vez generada la matriz A, debe generar el vector r_0 y debe proceder a calcular los nuevos r_i utilizando la relación de recurrencia que se presenta en (3). Finalmente, cuando se cumpla la propiedad (5), debe detener los cálculos e imprimir el vector r con los valores del PageRank.

Referencias

[1] LANGVILLE, A. N., AND MEYER, C. D. *Google's PageRank and beyond: The science of search engine rankings*. Princeton University Press, 2011.

[2] PAGE, L. Method for node ranking in a linked database, Sept. 4 2001. US Patent 6,285,999.