Algoritmo de Page Rank
Propuesta de paralelización

Por: Juan Camilo Rojas Cortés

Para la presente paralelización se trabajará teniendo en cuenta que, si se trabaja normalizando los valores del vector, Page Rank termina siendo una distribución de probabilidad en la que la suma de todos los valores calculados es 1. De hecho, este concepto se tiene en cuenta para la inicialización del sistema, tanto en el método iterativo como en el algebraico.

Dicho esto, se propone dividir el grafo inicial en segmentos de tamaño *n1*, de tal forma que si el este grafo inicial tiene *n* nodos, el grafo nuevo tendría *n/n1* nodos. Al hacer esta división se deberá tener en cuenta todas las aristas que entran y salen de los nodos iniciales que pertenecen a los grupos finales; siendo estos los arcos del grafo nuevo. Para ilustrar este concepto se procede a mostrar los siguientes gráficos:

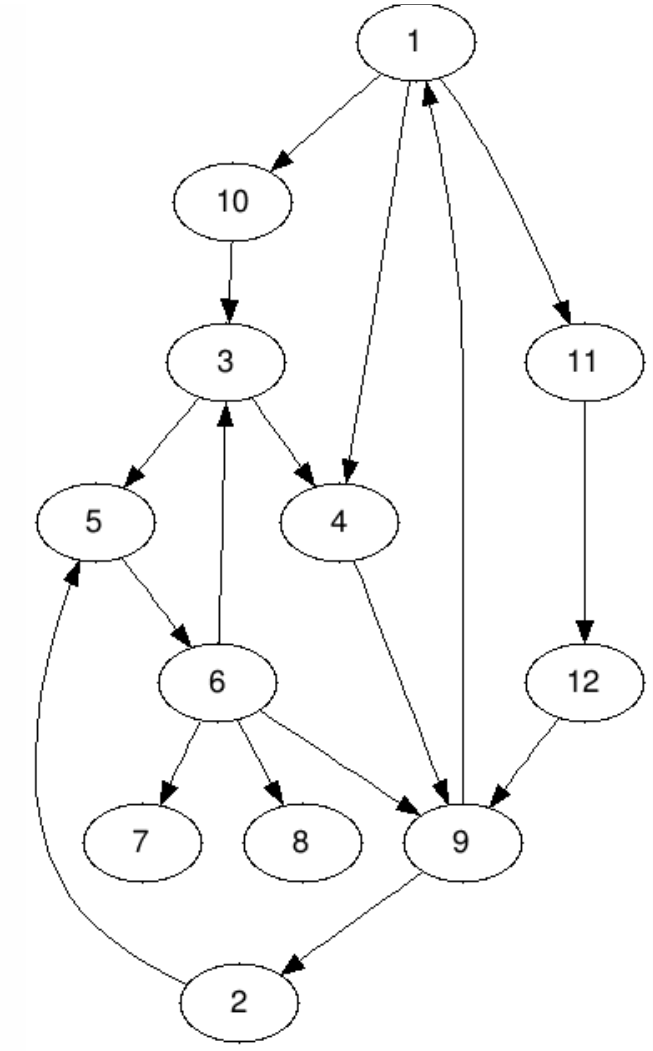

Figura 1. Grafo inicial con 12 vértices.

Del grafo representado en la figura 1 se elegirán 3 grupos aleatorios de 4 vértices. Para el presente ejemplo gráfico se elegirá la siguiente agrupación: Grupo A (1, 2, 3, 4), Grupo B (5, 6, 7, 8), Grupo C

(9, 10, 11, 12). Además, las aristas entrantes y salientes de cada uno de los grupos corresponderán a las aristas entrantes y salientes de los nodos iniciales que lo componen. A continuación, en la Figura 2 se ilustra la disposición del nuevo grafo.

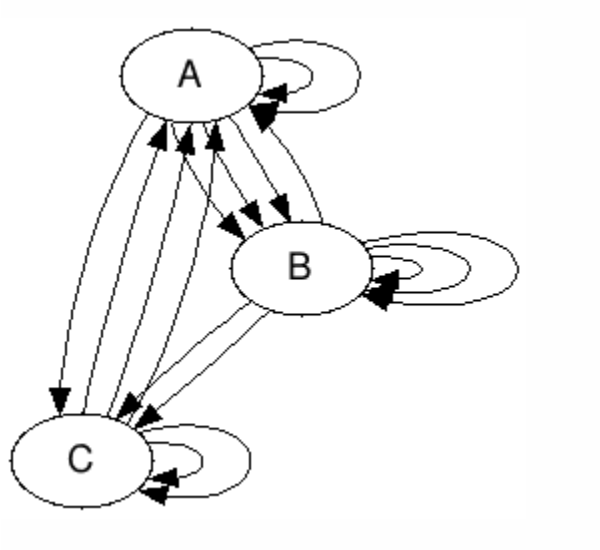

Figura 2. Grafo resultante de dividir el grafo de la Figura 1 en tres grupos (A, B y C).

Como se puede evidenciar, el grafo de la Figura 2 es mucho más simple para que se le realice Page Rank. Por lo tanto, una vez que se ha dividido, con el nuevo grafo se le calcula el Page Rank a cada uno de los grupos seleccionados (A, B y C). La suma de estos valores de Page Rank, como se explicó inicialmente, da 1 como resultado.

Después de haber calculado el Page Rank de los grupos, se descompone cada uno de estos en sus componentes originales formando un grafo por cada uno de ellos. En este grafo se van a tener en cuenta únicamente las aristas interiores entre sus nodos, sin incluir aquellas que salen o entran del grupo. Estas se omiten porque ya fueron tenidas en cuenta en el paso anterior. Matemáticamente, este asunto se resuelve una vez más con el concepto de probabilidad, pero haciéndole una ligera modificación: Esta vez la suma de estos page ranks pequeños no va a dar 1 como resultado, sino que debe dar el mismo valor que dio el page rank del grupo que lo compone.

Modelo de concurrencia elegido:

Para implementar la solución descrita anteriormente se decidió utilizar una arquitectura push/pull con fan, workers y sink. Se decidió utilizar esta arquitectura porque el hecho de que exista un fan facilita que se haga la repartición de la matriz de adyacencia de forma sencilla.

En ese orden de ideas, la secuencia de pasos a seguir sería la siguiente:

1. En el fan se divide el grafo inicial en grupos compuestos por n1 nodos, siendo n1 un parámetro establecido previamente.
2. En el fan se ejecutra page rank con el nuevo grafo generado. Se recomienda que el número de nodos por grupo sea grande, con el objetivo de que el cálculo de page rank en este paso sea rápido.

3. El fan genera n/n1 matrices de adyacencia nuevas (siendo n el número inicial de nodos y n1 el número de nodos por grupo) y envía cada una de estas a los workers que haya en el sistema. Además, envía como parámetro el page rank del grupo correspondiente.
4. Cada uno de los workers calcula page rank con el grupo asignado, teniendo en cuenta que la suma de estos valores debe dar como resultado el page rank del grupo, recibido previamente desde el fan.
5. Cada worker le envía el vector propio calculado al sink.
6. El sink concatena los vectores propios recibidos para así tener un aproximado del page rank de todos los nodos.