

# Tópicos Especiales en Telemática: Proyecto 3 – Clústering de Documentos a partir de Métricas de Similitud

Juan D. Pineda–Cárdenas

Edwin N. Montoya–Múnera

28 de septiembre de 2017

## Resumen

Este documento plantea el Proyecto3 de la materia Tópicos Especiales en Telemática. El proyecto 3 consiste en diseñar e implementar un algoritmo paralelo que permita agrupar (clustering) un conjunto de documentos utilizando el algoritmo de *k-means* y una métrica de similaridad entre documentos.

## 1. Objetivos

- Aplicar la metodología para el diseño de algoritmos paralelos (PCAM)
- Aplicar conocimientos de programación paralela usando estrategias adecuadas para resolver problemas intensivos en recursos computacionales.
- Analizar los resultados entre un acercamiento secuencial y paralelo incluyendo las distintas estrategias y tecnologías asociadas a Computación de Alto Rendimiento.
- Reducir tiempos de ejecución utilizando estrategias de desarrollo, tecnologías, herramientas y la infraestructura adecuada para ejecutar aplicaciones que requieran tiempos considerables y que de otra manera no sería posible ejecutarlos en un tiempo razonable.
- Entender las limitaciones a nivel de algoritmia, software y hardware asociadas a distintos problemas computacionales que podrían sortearse a partir de herramientas y estrategias como computación paralela y distribuida.
- Trabajar inter–disciplinariamente con otros profesionales y resolver problemas reales.

## 2. Enunciado del Problema

La minería de texto (*text mining*), es una de las técnicas de análisis de textos que ha permitido implementar una serie de aplicaciones muy novedosas hoy en día. Buscadores en la web (Google, Facebook, Amazon, Spotify, Netflix, entre otros), sistemas de recomendación, procesamiento natural del lenguaje, son algunas de las aplicaciones.

Las técnicas de agrupamiento de documentos (*clustering*) permiten relacionar un documento con otros parecidos de acuerdo a alguna métrica de similaridad. Esto es muy usado en diferentes aplicaciones como: Clasificación de nuevos documentos entrantes al dataset, búsqueda y recuperación de documentos, ya que cuando se encuentra un documento seleccionado de acuerdo al criterio de búsqueda, el contar con un grupo de documentos relacionados, permite ofrecerle al usuario otros documentos que potencialmente son de interés para él.

Se tomará como base el artículo: “Similarity Measures for Text Document Clustering” [Hua08], con el objetivo de realizar su paralelización.

## 2.1. Definición específica:

Se tiene un conjunto de documentos grande  $D$ , el cual contiene  $d_i$  documentos de texto, de algún corpus o *dataset* dado. En este caso, se puede utilizar alguno de estos *datasets*:

1. La base de textos de Gutenberg – <http://www.gutenberg.org>
2. El *dataset* dado por el artículo.

Las características que debe contener el dataset es que presente relaciones sintácticas y semántica entre los documentos, por ejemplo en un dataset de noticias presenta una correlación entre documentos de un mismo tipo de noticia (género). Una de las primeras actividades que deberán hacer los grupos de trabajo, es seleccionar adecuadamente el(los) dataset(s).

El problema básicamente se sintetiza en 2 subproblemas candidatos a ser paralelizables:

- **Subproblema 1:** Diseño e implementación de una función de similaridad entre 2 documentos, esto es: dato  $d_i$  y  $d_j$ , se define una función de de similaridad,  $fs$ , entre  $d_i$  y  $d_j$  como:  $fs(d_i, d_j) = [-min\_val, max\_val]$ . Hay varios algoritmos de similaridad [MRS08], de los cuales deberá elegir uno que se preste para ser paralelizado, entre los algoritmos más usados se encuentra: La distancia Euclidiana, Coseno, Jaccard o Pearson entre otros.
- **Subproblema 2:** Una vez se tiene definida e implementada la función de similaridad, se puede ejecutar el algoritmo de agrupamiento (*clustering*), que permite dividir el conjunto de documentos en un número de subgrupos. Estos algoritmos de *clustering* pertenecen a la categoría de Aprendizaje de Máquina NO supervisado. Uno de los algoritmos más usados es *k-means*, el cual permite dividir el conjunto de documentos en  $k$  subgrupos. Este algoritmo requiere conocer de antemando  $k$ , por lo cual, está dentro del alcance de este proyecto<sup>3</sup>, realizar varias simulaciones con  $k$  diferentes, el valor de  $k$  es calculado de forma heurística, teniendo en cuenta algunas de las siguientes consideraciones: 1) variando apriori  $k$  en algún intervalo dado: ej:  $3 \leq k \leq 5$ . 2) conociendo en mayor detalle el dataset, lo que permita inferir un  $k$ , por ejemplo en un conjunto de noticias, se podría sugerir un valor central de  $k$  alrededor del número de géneros presentes de tipos de noticias, ej: 10 géneros de noticias,  $8 \leq k \leq 12$ .

Este problema se puede atacar de varias maneras, sin embargo es necesario establecer un punto de partida. El primer acercamiento es analizar la similaridad de cada documento con los demás documentos por medio de un algoritmo serial. Este algoritmo establecerá la línea base para poder comparar las aceleraciones obtenidas a partir de las estrategias de cómputo paralelo en términos de tiempo de ejecución y recursos consumidos. Llegados a este punto, se establece como se deber realizar el particionamiento funcional y/o por datos. Tenga en cuenta que usted deberá no solo hallar la distancia entre los documentos sino deberá encontrar los distintos *clusters* de los subgrupos de documentos, es decir, sus resultados deberán diferenciar a que *cluster* pertenece cada documento a partir de su similaridad.

La idea es que usted diseñe una solución donde se reduzcan los tiempos para el análisis de cada uno de los documentos con respecto a los demás documentos y poder hacer un comparativo teniendo como línea base el tiempo de procesamiento de este problema en un acercamiento secuencial y centralizado con respecto a uno paralelo y distribuido, haciendo un análisis de las herramientas, la estrategia, hallando la aceleración del algoritmo secuencial con respecto al paralelo y con respecto al distribuido.

## 3. Restricciones y Condiciones

A continuación algunas de las exigencias y condiciones que se requieren en el trabajo final.

- Procesar todos los documentos del dataset utilizado. Si es Gutenberg, acá puede encontrar una muestra del dataset:<sup>1</sup> o los *datasets* enunciados en el paper, tenga en cuenta que a partir de esto usted deberá realizar un análisis comparativo en el cual tendrá una tabla de tiempos de implementaciones vs. *datasets*.

---

<sup>1</sup><https://goo.gl/LL4CgA>

- Aplicar la metodología PCAM
- Utilizar el paradigma de paso de mensajes MPI (opcionalmente puede complementar el procesamiento MPI con multiprocesamiento local o coprocesamiento)
- Establecer los tipos de datos a usar, la estrategia de paralelización y la tecnología a utilizar.
- Tenga en cuenta que esta deberá ser una aplicación no interactiva, ya que lo que se requiere es poder usarla en un clúster MPI.
- Documentar el punto anterior justificando las decisiones tomadas
- La documentación y código generado deberá ser compartido con el profesor por medio de Github<sup>2</sup>
- Debe escribir el informe de resultados del proyecto en formato artículo, considerando, pero no limitada a estas secciones del artículo:
  1. Título
  2. Afiliaciones (nombre, correo electrónico, institución)
  3. Palabras clave
  4. Resumen
  5. Introducción
  6. Marco Teórico (Descripción del problema)
  7. Análisis y Diseño mediante PCAM incluyendo algoritmos y estructuras de datos y rendimiento analítico de la solución.
  8. Implementación
  9. Análisis de resultados (secuencial vs. paralelo y resultados de aceleraciones entre *datasets* e implementaciones en los cuales deberá incluir gráficos de aceleración y distintas ejecuciones.)
  10. Conclusiones
  11. Referencias
  12. Anexos (opcional)

## 4. Herramientas

Se recomienda el uso de las siguientes herramientas:

- Python 2.7 o 3, C o C++
- MPI4Py <http://pythonhosted.org/mpi4py/>, OpenMP, OpenMPI o MPICH2
- Cluster de MPI del Datacenter Académico
- Github
- L<sup>A</sup>T<sub>E</sub>X
- PyCUDA si considera que el problema aplica para este tipo de arquitectura
- Makefile

---

<sup>2</sup><https://github.com/>

## 5. Criterios de Evaluación

- **40 % Reporte Técnico:** El reporte técnico (RT) deberá estar escrito en formato de artículo científico. El reporte técnico, deberá estar entre 6-10 páginas y deberá estar escrito usando la plantilla IEEE o ACM para L<sup>A</sup>T<sub>E</sub>X para artículos científicos.
- **60 % Implementación:**
  - **40 % Completitud** (funcional y eficiencia)
  - **10 % Legibilidad**
  - **10 % Documentación de usuario** (Colocada en el repo en formato markdown)

Tanto el Reporte Técnico como la implementación hacen parte de la sustentación. Cada ítem tendrá un factor multiplicativo de 0 a 1, el cual será determinado durante la sustentación.

LA NOTA MÍNIMA APROBATORIA CORRESPONDIENTE A 3.0, CORRESPONDE A LA ENTREGA CORRECTA DEL REPORTE TÉCNICO, LA EJECUCIÓN CORRECTA FUNCIONAL Y PARALELA DEL SISTEMA Y HACE LA SUSTENTACIÓN ADECUADA. ENTRE 3.0 Y 5.0 PARA LA CALIDAD, EFICIENCIA Y SUSTENTACIÓN DEL PROYECTO. MENOR QUE 3.0 CUANDO NO FUNCIONA EN FORMA PARALELA O REPORTE TÉCNICO INCOMPLETO/REGULAR O NO SE EVIDENCIA APROPIACIÓN DEL PROYECTO EN LA SUSTENTACIÓN.

LA NOTA INDIVIDUAL POR ALUMNO: 50 % PROYECTO, 25 % SUSTENTACIÓN, 25 % PROMEDIO DE LAS SUSTENTACIONES DEL GRUPO.

## 6. Fechas Importantes

- **Domingo 15 de Octubre 11:59 p.m.** – Entrega de la práctica
- **Domingo 22 de Octubre 11:59 p.m.** – Entrega del Reporte Técnico
- **Lunes 23 al Miércoles 25 de Octubre** – Sustentación de la práctica

## 7. Código de Honor

- En la documentación del proyecto (README.md y Reporte Técnico), cada grupo reconocerá los créditos y autoría de componentes reutilizados de otros proyectos a nivel de código fuente, documentación (correcta citación), diseños o algoritmos.
- Cada miembro de equipo, firmará el Reporte Técnico y README.md mencionando explícitamente que el trabajo es auténtico, original, no copiado, no enviado a realizar por un tercero, y que reconoce a los terceros que aportaron al proyecto directa o indirectamente.

## Referencias

- [Hua08] Anna Huang. Similarity measures for text document clustering. *Proceedings of the Sixth New Zealand*, (April):49–56, 2008.
- [MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.