Departamento:
DEPARTAMENTO DE INGENIERÍA E INVESTIGACIONES TECNOLÓGICAS

Programa de acreditación:
PROINCE

Programa de Investigación[1]:

Código del Proyecto:
C225
Título del proyecto
Resolución Eficiente de la Descomposición en Valores Singulares en una Arquitectura Híbrida y su Posterior Inserción en un Sistema de Recuperación de Información

PIDC: ☐
Elija un elemento.
PII: ☐
Elija un elemento.
Director:
RYCKEBOER, Hugo Emilio
Director externo:

Codirector:
SPOSITTO, Osvaldo Mario
Integrantes:
CORA, Gabriela
LEDESMA, Viviana

Investigador Externo, Asesor- Especialista, Graduado UNLaM:

Alumnos de grado: (Aclarar si tiene Beca UNLaM/CIN)
PROCOPIO, Gastón Emanuel
QUINTANA, Fabio
Alumnos de posgrado:
Resolución Rectoral de acreditación: N°
392/2019
Fecha de inicio:
01/01/2019
Fecha de finalización:
30/04/2020

[1] Los Programas de Investigación de la UNLaM están acreditados con resolución rectoral, según lo indica la Resolución HCS Nº 014/15 sobre **Lineamientos generales para el establecimiento, desarrollo y gestión de Programas de Investigación a desarrollarse en la Universidad Nacional de La Matanza**. Consultar en el departamento académico correspondiente la inscripción del proyecto en un Programa acreditado.

A. Desarrollo del proyecto (adjuntar el protocolo)

A.1. Grado de ejecución de los objetivos inicialmente planteados, modificaciones o ampliaciones u obstáculos encontrados para su realización (desarrolle en no más de dos (2) páginas):

Se han cumplido con las actividades previstas en el cronograma, alcanzando los siguientes objetivos específicos:

- Búsqueda bibliográfica y estudio de diferentes métodos para resolver el cálculo de la DVS de una matriz real densa.
- Búsqueda bibliográfica y estudio de la arquitectura a bajo nivel de la CPU Intel I7 7700
- Búsqueda bibliográfica y estudio de la arquitectura a bajo nivel de la GPU Geforce GTX 1080ti
- Implantación de algoritmos en CPU
- Implantación de algoritmos en CPU gobernando explícitamente los multi-núcleos
- Implantación de algoritmos en GPU
- Implantación Híbrida 1 GPU + 1 CPU
- Análisis de los resultados de las distintas implementaciones

Ampliando lo realizado y listado anteriormente, se hizo una revisión bibliográfica, enfocada principalmente a descubrir distintos métodos para resolver la DVS. En particular, este proyecto se enfoca en los algoritmos basados en bidiagonalización, los cuales aplican transformaciones ortogonales con el fin de obtener una forma bidiagonal para luego conseguir la DVS de la matriz bidiagonal, en general los distintos métodos utilizan para sus cálculos las transformaciones de Householder que se aplican por la derecha y por la izquierda de la matriz. En base al material bibliográfico revisado y al análisis de distintos métodos se decidió poner especial interés en dos algoritmos alternativos de bidiagonalización que están pensados para soportar el paralelismo, un propuesto por Ralha[2] y el otro por Barlow[3]. En estas propuestas la bidiagonalización es unilateral, las transformaciones de Householder son aplicadas solamente por el lado derecho de la matriz. La elección de dichos algoritmos se fundamenta en que la meta establecida para este proyecto se relaciona con la paralelización de los mismos en distintas arquitecturas, y en particular en aquellas basadas en GPU.

En una primera etapa del proyecto se implementó un algoritmo para bidiagonalización, de tipo secuencial, que utiliza transformaciones de Householder por la izquierda y por la derecha de la matriz, en el lenguaje de programación C#, se utilizó para ello el entorno Visual Studio 2015 con el complemento NVIDIA® Nsight™. El principal aporte de esta actividad tiene que ver con que este algoritmo ofrece evidencia de las funciones internas del proceso de bidiagonalización. Esta parte del trabajo ha quedado parcialmente publicado en el artículo [1] "Aplicación de la Descomposición de Valores Singulares a un Sistema de Recuperación de Información", publicado en la revista REDDI.

Por otra parte, se realizó un estudio a bajo nivel de la arquitectura de la GPU y su utilización para operaciones matemáticas y algebraicas de forma paralelizada. Las GPU disponibles para este trabajo

[2] Ralha, R. "One-sided reduction to bidiagonal form". *Linear Algebra and Its Applications, ELSEVIER*, 358(1-3): 219-238, 2003.
[3] Barlow, J., Bosner, N., Drmač, Z. "A new stable bidiagonal reduction algorithm". *Linear Algebra and Its Applications, ELSEVIER*, 397: 35-84, 2005.

de investigación fueron: Nvidia G210, Geforce Gtx 650 y Ati radeon R9 390x. En cuanto a esta parte de la investigación se realizó un escrito para capacitación de todos los integrantes del equipo. Se ha implementado un algoritmo que paraleliza el método de Householder, procedente de un trabajo previo[4] realizado por integrantes de este proyecto de investigación, el que permitió experimentar con la arquitectura basada en GPU. Quedó confirmado que las diferencias entre las arquitecturas tienen un rol importante en la resolución de los cálculos, durante las pruebas se puso en evidencia que determinadas operaciones resultarán convenientes para realizar en CPU dependiendo del tamaño de las matrices involucradas, ya que debe considerarse los tiempos de envío y recepción de información entre la memoria RAM y la GPU, y viceversa.

Seguidamente se desarrolló en C# el algoritmo de bidiagonalización propuesto por Barlow, el cual, como se mencionó antes, aplica Householder por un solo lado. Se realizaron pruebas con matrices cuadradas de distintas dimensiones y con valores positivos, los resultados obtenidos al aplicar la bidiagonalización fueron contrastados en tres arquitecturas diferentes CPU Monoprocesador, CPU Multiprocesador y GPU. Los resultados permitieron detectar los casos en que resultaba ventajoso ejecutar este algoritmo en CPU monoprocesador, para matrices de dimensiones inferiores a un orden de 200, en cambio, para aquellas de mayor dimensión una arquitectura basada GPU mejoraba notablemente los tiempos de respuesta. Mas detalles de este estudio fueron publicados en el trabajo [2] "Comparación de un Algoritmo de Bidiagonalización para su Utilización en la Recuperación de Información", presentado en el CACIC 2020.

Seguidamente, se avanzó hacia una arquitectura híbrida CPU-GPU a fin de analizar si es posible obtener mejoras en los tiempos insumidos con respecto a la ejecución completa en GPU. Esto implicó, entre otras cosas, identificar cuál sería el punto de quiebre o cota, para definir qué columnas de la matriz se deberían procesar en la GPU, dejando las restantes para ser ejecutadas en la CPU. Los valores obtenidos con la implementación en esta arquitectura híbrida fueron contrastados con la ejecución completa en GPU, parte de este estudio fue publicado en el trabajo [3] "Implantación de un Algoritmo de Bidiagonalización en un Entorno Híbrido para su Aplicación en la Recuperación de Información" presentado en CONAIISI 2020.

En paralelo a lo anterior también se ha programado el método de Ralha. Se realizaron pruebas en matrices cuadradas de distintas dimensiones, obteniendo resultados con diferencias ínfimas respecto del método de bidiagonalización de Barlow (diferencias nulas o del orden de 10^-15). Se realizó la paralelización del algoritmo utilizando los múltiples núcleos de la CPU obteniendo mejoras en los tiempos para obtener un resultado. Para citar un ejemplo, se pasó 19,5 segundos a unos 6,7 segundos en matrices de 1000x1000 elementos.

Con respecto a la tarea que se había planificado para la implantación de los algoritmos en CPU conectados en red, hubo que postergarla, por razones de público conocimiento, debido a la imposibilidad de realizar las pruebas presencialmente en el laboratorio. Así también, debido a la reducción del equipo, por la baja de dos integrantes, no pudo ser cumplimentada la actividad referida a la ampliación del corpus documental.

En cuanto a la inserción del algoritmo en el SRI, se requirió la adecuación del sistema de recuperación para reemplazar el algoritmo de bidiagonalización secuencial original por el algoritmo de Barlow paralelizado y adaptado para una arquitectura basada en GPU que fue descrito en el trabajo [2].

[4] Spositto, O., Procopio, G., Quintana F., Ryckeboer, H. "Una paralelización del método de Householder". *XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016).*

B. Principales resultados de la investigación

B.1. Publicaciones en revistas (informar cada producción por separado)

Artículo 1:	
Autores	Osvaldo Spositto Viviana Ledesma Gastón Procopio
Título del artículo	Aplicación de la Descomposición de Valores Singulares a un Sistema de Recuperación de Información
N° de fascículo	*4*
N° de Volumen	*2*
Revista	*REDDI*
Año	*2019*
Institución editora de la revista	*UNLaM*
País de procedencia de institución editora	*Argentina*
Arbitraje	SI
ISSN:	2525-1333
URL de descarga del artículo	https://reddi.unlam.edu.ar/index.php/ReDDi/article/view/91
N° DOI	

B.2. Libros

Libro 1	
Autores	
Título del Libro	
Año	
Editorial	
Lugar de impresión	
Arbitraje	Elija un elemento.
ISBN:	
URL de descarga del libro	
N° DOI	

B.3. Capítulos de libros

Autores	
Título del Capitulo	
Título del Libro	
Año	

Editores del libro/Compiladores	
Lugar de impresión	
Arbitraje	Elija un elemento.
ISBN:	
URL de descarga del capítulo	
N° DOI	

B.4. Trabajos presentados a congresos y/o seminarios

Trabajo 1	
Autores	Sposito, Osvaldo Mario Ledesma, Viviana Procopio, Gastón Ryckeboer, Hugo Emilio
Título	Hacia la optimización de un sistema de recuperación de información
Año	2020
Evento	XXII Workshop de Investigadores en Ciencias de la Computación (WICC)
Lugar de realización	UNPA, El Calafate, Santa Cruz
Fecha de presentación de la ponencia	Mayo 2020
Entidad que organiza	Red de Universidades con Carreras en Informática (RedUNCI)
URL de descarga del trabajo (especificar solo si es la descarga del trabajo; formatos pdf, e-pub, etc.)	http://sedici.unlp.edu.ar/handle/10915/104116

Trabajo 2	
Autores	Osvaldo Sposito, Viviana Ledesma, Gastón Procopio, Hugo Ryckeboer, Victoria Saizar y Alexis Vainberg
Título	Comparación de un Algoritmo de Bidiagonalización para su Utilización en la Recuperación de Información
Año	2020
Evento	XXVI Congreso Argentino de Ciencias de la Computación (CACIC 2020)
Lugar de realización	UNLaM, Bs. As.
Fecha de presentación de la ponencia	5 AL 9 de octubre de 2020
Entidad que organiza	Red de Universidades con Carreras en Informática (RedUNCI)
URL de descarga del trabajo (especificar solo si es la descarga del trabajo; formatos pdf, e-pub, etc.)	https://cacic2020.unlam.edu.ar/es-ar/pdf/2020-CACIC-LIBROACTAS-3.pdf

Trabajo 3	
Autores	Osvaldo Sositto, Viviana Ledesma, Gastón Procopio, Victoria Saizar y Alexis Vainberg
Título	Implantación de un Algoritmo de Bidiagonalización en un Entorno Híbrido para su Aplicación en la Recuperación de Información
Año	2020
Evento	8vo.Congreso Nacional de Ingeniería Informática y Sistemas de Información (CoNaIISI 2020)
Lugar de realización	UTN Facultad Regional San Francisco, Córdoba.
Fecha de presentación de la ponencia	5 AL 9 de octubre de 2020
Entidad que organiza	Red RIISIC
URL de descarga del trabajo (especificar solo si es la descarga del trabajo; formatos pdf, e-pub, etc.)	

B.5. Otras publicaciones

Autores	
Año	
Título	
Medio de Publicación	

C. Otros resultados. Indicar aquellos resultados pasibles de ser protegidos a través de instrumentos de propiedad intelectual, como patentes, derechos de autor, derechos de obtentor, etc. y desarrollos que no pueden ser protegidos por instrumentos de propiedad intelectual, como las tecnologías organizacionales y otros. Complete un cuadro por cada uno de estos dos tipos de productos.

C.1. Títulos de propiedad intelectual. Indicar: Tipo (marcas, patentes, modelos y diseños, la transferencia tecnológica) de desarrollo o producto, Titular, Fecha de solicitud, Fecha de otorgamiento

Tipo	Titular	Fecha de Solicitud	Fecha de Emisión

C.2. Otros desarrollos no pasibles de ser protegidos por títulos de propiedad intelectual. Indicar: Producto y Descripción.

Producto	Descripción

D. Formación de recursos humanos. Trabajos finales de graduación, tesis de grado y posgrado. Completar un cuadro por cada uno de los trabajos generados en el marco del proyecto.

D.1. Tesis de grado

Director (apellido y nombre)	Autor (apellido y nombre)	Institución	Calificación	Fecha /En curso	Título de la tesis

D.2 Trabajo Final de Especialización

Director (apellido y nombre)	Autor (apellido y nombre)	Institución	Calificación	Fecha /En curso	Título del Trabajo Final

D.2. Tesis de posgrado: Maestría

Director (apellido y nombre)	Tesista (apellido y nombre)	Institución	Calificación	Fecha /En curso	Título de la tesis
Osvaldo Spositto	Mauro Javier Casuscelli	UNLaM	-	En Curso	Estudio comparativo de DBSCAN, KMEANS con redes neuronales en un Sistema de Recuperación de Información

D.3. Tesis de posgrado: Doctorado

Director (apellido y nombre)	Tesista (apellido y nombre)	Institución	Calificación	Fecha /En curso	Título de la tesis

D.4. Trabajos de Posdoctorado

Director (apellido y nombre)	Posdoctorando (apellido y nombre)	Institución	Calificación	Fecha /En curso	Título del trabajo	Publicación

E. Otros recursos humanos en formación: estudiantes/ investigadores (grado/posgrado/ posdoctorado)

Apellido y nombre del Recurso Humano	Tipo	Institución	Período (desde/hasta)	Actividad asignada[5]
Gastón Procopio	Estudiante de grado	UNLaM	1/1/2019 – 30/04/2021	Programación de Algoritmos
Fabio Quintana	Estudiante de grado	UNLaM	1/1/2019 – 30/04/2021	Programación de Algoritmos

F. Vinculación[6]: Indicar conformación de redes, intercambio científico, etc. con otros grupos de investigación; con el ámbito productivo o con entidades públicas. Desarrolle en no más de dos (2) páginas.

En este período se ha mantenido vinculación mediante reuniones con otro equipo de investigación dependiente también del Departamento de Ingeniería e Investigaciones Tecnológicas de la UNLaM, el cual es dirigido por Graciela De Luca, dicho equipo ha estado trabajando en una línea de investigación relacionada a la utilización de tecnologías adaptativas para la optimización del uso de recursos y eficiencia energética en clusters de servidores GPU y CPU, así también, en la aplicación de GP-GPU Computing para la optimización de algoritmos científicos mediante el uso de profiling de hardware. Dicho intercambio fue útil para familiarizarse con distintas bibliotecas de optimización de programación que utilizan GPU.

G. Otra información. Incluir toda otra información que se considere pertinente.

<u>Actividades de Difusión en Eventos Científicos</u>:

- Presentación de pósteres en la mesa de trabajo de Ingeniería de Software: Ledesma. Títulos de Artículos: “Hacia la optimización de un sistema de recuperación de información”. Evento: WICC 2020 – XX Workshop de Investigadores en Ciencias de la Computación. Lugar: Virtual, organizado por la UNPA, El Calafate, Santa Cruz. Fecha: abril 2020.
- Expositores: Ledesma y Procopio. Título del Artículo: “Comparación de un Algoritmo de Bidiagonalización para su Utilización en la Recuperación de Información”. Evento: XXVI Congreso Argentino de Ciencias de la Computación

[5] Descripción de la/s actividad/es a cargo (máximo 30 palabras)

[6] Entendemos por acciones de “vinculación” aquellas que tienen por objetivo dar respuesta a problemas, generando la creación de productos o servicios innovadores y confeccionados “a medida” de sus contrapartes.

(CACIC 2020) Lugar: Virtual, organizado por la UNLaM, Bs. As. Fecha: agosto 2020.

- Expositora: Ledesma. Título del Artículo: "Implantación de un Algoritmo de Bidiagonalización en un Entorno Híbrido para su Aplicación en la Recuperación de Información". Evento: 8vo.Congreso Nacional de Ingeniería Informática y Sistemas de Información (CoNaIISI 2020). Lugar: Virtual, organizado por la UTN, Facultad Regional San Francisco, Córdoba. Fecha: octubre 2020.

Modificaciones a la Composición del Equipo:

- El Mg. Julio Bossero fue dado de baja del proyecto al renunciar a partir del 31 de diciembre de 2019, debido a su falta de disponibilidad horaria por iniciar otra actividad en el ámbito profesional.

- El Ing. Mauro Casuscelli fue dado de baja del proyecto a partir de su renuncia, desde el 1 de marzo de 2020, por no contar con disponibilidad de tiempo para continuar con las actividades del proyecto.

Formación de Recursos Humanos:

- En noviembre de 2019 la Ing. Viviana Ledesma presentó su tesis de maestría y su posterior defensa, el título de la misma fue "Estrategia de Requisitos adaptable según factores de situación", la cual desarrolló en la UNLaM dirigida por la Dra. Graciela Hadad y codirigida por el Ing. Jorge Horacio Doorn.
- El alumno Gastón Procopio, finalizó su carrera de Ingeniería en Informática, en diciembre de 2019, con la aprobación del proyecto final de carrera referido a un sistema de seguimiento de voz para pacientes enfermos de Parkinson.

Por otra parte, los integrantes del proyecto han realizado los siguientes cursos o asistido a jornadas de capacitación durante el transcurso del proyecto:

- Ledesma - Jornada: "Comunicación Científica: Investigar y Publicar" organizada por la Secretaría de Ciencia y Tecnología en conjunto con la Asociación de Docentes de la UNLaM. Duración: 3 hs. Fecha: 25 de junio 2019.
- Ledesma – Jornada: "Capacitación MS Teams para Docentes UNLaM". Modalidad Virtual. Organizado por la Secretaría Académica de la UNLaM. Fecha: 21 de abril de 2020.
- Ledesma – Encuentro: "Encuentro para fortalecimiento de los aspectos pedagógicos de la cursada a distancia". Modalidad virtual. Organizado por el Departamento de Ingeniería e Investigaciones Tecnológicas de la UNLaM. Duración: 2 hs. Fecha: 29 de abril de 2020.
- Procopio – Curso: "Metodologías Ágiles por Rus". Organizado por RUS Seguros. Duración: 9 hs. Fecha: 5, 12 y 19 de agosto de 2020.
- Quintana – Curso: "Competencia y Desarrollo del Talento en la Nube para la Comunidad de la UNLaM". Organizado por AWS Academy Cloud Foundations. Modalidad virtual. Organizado por Amazon y UNLaM. Duración: 14 hs. Fecha: 9 de noviembre a 21 de diciembre de 2020.

- Ledesma – Curso: "Inteligencia Artificial y Derecho". Modalidad Virtual. Organizado por elDial.cursos. Duración: 5 hs. Fecha: 21 de diciembre de 2020.

Actividades de Evaluación:

- Spositto: Jurado de tesis de maestría de la alumna Viviana Ledesma. Título: Estrategia de Requisitos adaptable según factores de situación. Carrera: Maestría en Informática. Escuela de Posgrado, Universidad Nacional de La Matanza. Aprobada. Fecha: 26/11/2019.
- Ledesma: Evaluación de capítulos de enciclopedia: Encyclopedia of Organizational Knowledge, Administration, and Technologies. First Edition. IGI GLOBAL, INTERNATIONAL ACADEMIC PUBLISHER. Agosto 2019.
- Ledesma: Miembro del Comité de Programa con evaluación de artículos. Evento: 23rd Workshop on Requirements Engineering (WER 2020). Agosto de 2020.

Otras Actividades Científicas y Tecnológicas:

- Ryckeboer y Bossero: han participado como miembros del Comité Académico en la categoría Base de Datos, Artículos de Investigación, en el VII Congreso Nacional de Ingeniería Informática – Sistemas de Información, CoNaIISI 2019, realizado los días 14 y 15 de noviembre en la Universidad Nacional de La Matanza.
- Cora y Ledesma: han participado como Coordinadoras de Sesión en la categoría Base de Datos, Artículos de Investigación, en el VII Congreso Nacional de Ingeniería Informática – Sistemas de Información, CoNaIISI 2019, realizado los días 14 y 15 de noviembre en la Universidad Nacional de La Matanza
- Quintana: ha sido miembro del Comité Administrativo del evento CoNaIISI 2019, organizado por la UNLaM y realizado en la UNLaM los días 14 y 15 de noviembre de 2019.
- Quintana: ha sido miembro del Comité Administrativo del evento CACIC 2020, organizado por la UNLaM y realizado en forma virtual del 5 al 9 de octubre de 2020.
- Quintana: ha colaborado brindando asistencia técnica para la retransmisión del evento EXPO PROYECTO 2020, organizado por la UNLaM y realizado en forma virtual del 9 al 13 de noviembre de 2020.

H. Cuerpo de anexos:

- Anexo I: Copia de cada uno de los trabajos mencionados en los puntos B, C y D, y certificaciones cuando corresponda.[7]
- Anexo II:
  - FPI-013: Evaluación de alumnos integrantes. (si corresponde)
  - FPI-014: Comprobante de liquidación y rendición de viáticos. (si corresponde)

[7] En caso de libros, podrá presentarse una fotocopia de la primera hoja significativa o su equivalente y el índice.

  - FPI-015: Rendición de gastos del proyecto de investigación acompañado de las hojas foliadas con los comprobantes de gastos.
  - FPI-035: Formulario de reasignación de fondos en Presupuesto.
- Anexo III: Alta patrimonial de los bienes adquiridos con presupuesto del proyecto (FPI 017)
- Nota justificando baja de integrantes del equipo de investigación.

_____

Firma y aclaración
del director del proyecto.

Lugar y fecha: 28 de febrero de 2021.

ANEXO I

COPIA DE ARTÍCULOS

Revista Digital del Departamento de Ingeniería e Investigaciones Tecnológicas de la Universidad Nacional de La Matanza

ISSN: 2525-1333
Vol.:4-Nro.2 (Diciembre-2019)

comunicación corta

Aplicación de la Descomposición de Valores Singulares a un Sistema de Recuperación de Información

Singular Value Decomposition Applied to Information Retrieval System

Osvaldo Sositto [(1)], *Viviana Ledesma* [(2)], *Gastón Procopio* [(3)]

[(1)] Universidad Nacional de La Matanza
sositto@unlam.edu.ar

[(2)] Universidad Nacional de La Matanza
vledesma@unlam.edu.ar

[(3)] Universidad Nacional de La Matanza
gprocopio@unlam.edu.ar

Resumen:

ISSN: 2525-1333. Vol.:4-Nro.2 (Diciembre-2019)

Este artículo se realiza en el marco de una investigación que tiene por objetivo optimizar un Sistema de Recuperación de Información, de desarrollo propio, mediante implementar y evaluar distintos algoritmos secuenciales y paralelos para resolver eficientemente la Descomposición de Valores Singulares. Dicho proceso comienza con la reducción de la matriz inicial a la forma bidiagonal. Estudios demuestran que la bidiagonalización puede consumir más del 70% del tiempo total del proceso. Por ello, como trabajo preliminar se han estudiado distintos métodos de bidiagonalización y se ha implementado un algoritmo basado en las transformaciones de Householder. El mismo se ha planteado con la suficiente flexibilidad como para ser adaptado fácilmente a otros algoritmos alternativos con el fin de realizar futuras implementaciones en arquitecturas paralelas, en particular las basadas en unidades de procesamiento gráfico.

Abstract:

This article is written in the context of an investigation whose objective is to optimize an information retrieval system in-house developed. It is done through the implementation and evaluation of different sequential and parallel algorithms to efficiently resolve the singular value decomposition. This process begins with the bidiagonal reduction of the initial matrix. Studies show that the bidiagonalization process can consume more than 70% of the overall time. Consequently, as groundwork, different bidiagonalization methods have been studied and one specific algorithm based on Householder transformations has been implemented. The latter has been set out with enough flexibility to be easily adapted to alternative algorithms in order to make future implementations in parallel architectures, particularly those based on graphic processing units.

Palabras Clave: *Descomposición de Valores Singulares, Bidiagonalización, Sistemas de Recuperación de Información*

Key Words: *Singular Value Decomposition*, *Bidiagonalization, Retrieval Information System*

Colaboradores: *Fabio QUINTANA, Victoria SAIZAR, Alexis VAINBERG*

I. CONTEXTO

Este artículo se enmarca en una línea de investigación, relacionada a los Sistemas de Recuperación de Información (SRI) realizada por investigadores del Departamento de Ingeniería e Investigaciones Tecnológicas de la Universidad Nacional de La Matanza. Particularmente se asocia al proyecto PROINCE, código C225, *Resolución Eficiente de la Descomposición en Valores Singulares en una Arquitectura Híbrida y su Posterior Inserción en un Sistema de Recuperación de Información*, con vigencia 2019-2020.

II. INTRODUCCIÓN

Desde un punto de vista práctico la Descomposición de Valores Singulares (DVS) tiene diferentes aplicaciones, compresión de imágenes digitales, reconocimiento facial, sistemas de recomendación, la indexación semántica latente (LSI, por sus siglas en inglés), entre otros [1].

Dado que el contexto de esta investigación se relaciona a un SRI desarrollado por el propio equipo, la LSI resulta de particular interés. Se trata de un método para la búsqueda de información en documentos a través de la indexación de términos [2]. Con la LSI se pretende la resolución de perturbaciones en la recuperación de información debido a problemas de sinonimia y polisemia o equivocidad del habla corriente. Por ejemplo, si se desea buscar la palabra “estación”, la cual tiene múltiples significados (polisemia) una búsqueda literal de la palabra produciría muchos resultados posibles (estación de tren, estación del año, etc.). Si lo que se desea buscar es “estación del año”, podrían interesar resultados de palabras distintas, pero con un significado igual o similar, por ejemplo “temporada”, “época” y así por el estilo (sinonimia). La LSI permite la búsqueda por conceptos o definiciones (en contraposición a la búsqueda literal).

Con tal objetivo se aplican algoritmos matemáticos especializados, que como resultado simulan el análisis que realizaría una persona. Una técnica ampliamente utilizada a tal fin es la DVS, luego la recuperación se realiza utilizando como punto de partida los valores singulares y vectores obtenidos a partir de la aplicación de dicha técnica [3].

Mediante este proyecto se pretende optimizar la resolución de la DVS, en especial mediante implementar algoritmos para resolver la primera fase de este proceso, la bidiagonalización. El resultado final de este proyecto se orienta hacia algoritmos que puedan ser implementados en plataformas paralelas y, en particular aprovechando la capacidad de las unidades de procesamiento gráfico (GPU, por sus siglas en inglés). En principio fue necesario evaluar distintos métodos de bidiagonalización y, a modo inicial, se implementó un algoritmo genérico, secuencial, que sirvió para evidenciar el funcionamiento interno del proceso.

III. MÉTODOS

La metodología utilizada para cumplir con los objetivos de este proyecto se llevará adelante realizando los siguientes pasos:

- Revisión en la literatura sobre los fundamentos matemáticos de la DVS y sus posibles variantes.
- Estudio de las tecnologías existentes para la implementación de la programación en paralelo que utiliza GPU.
- Análisis de las librerías paralelas: CUDA (Compute Unified Device Architecture) y CuBlas (CUDA Basic Linear Algebra Subprograms) para guiar en la instalación, introducción a la arquitectura, desarrollo y ejecución de programas que utilicen la tecnología BLAS (Basic Linear Algebra Subprograms) y

LAPACK (Linear Algebra PACKage), como herramienta de apoyo para Computación de Altas Prestaciones.

- Desarrollo de librerías propias en lenguaje de programación C#.

IV. RESULTADOS Y OBJETIVOS

En esta etapa preliminar del proyecto se ha implementado un algoritmo en el lenguaje de programación C# para la bidiagonalización basada en transformaciones de Householder. Si bien existía la posibilidad de utilizar la biblioteca LAPACK[1], la documentación y las rutinas utilizadas en este paquete resultan más difíciles de entender y requieren más tiempo para dominarse. En cambio, el disponer del código en C# ofrece como ventaja, por una parte, permitir la comprensión de cada etapa interna del proceso, y por otra sienta las bases para que este código posteriormente pueda ser adaptado a diferentes algoritmos de bidiagonalización, e implementarlos en otras arquitecturas paralelas, en particular aquellas basadas en GPU, a fin de analizar su eficiencia.

Los próximos objetivos por cumplir para este proyecto son los siguientes:

- Implementar los algoritmos alternativos propuestos por Ralha y por Barlow, en una arquitectura basada en CPU.
- Adaptar los mismos algoritmos para ser implementados en una arquitectura basada en GPU.
- Realizar un estudio comparativo en cuanto al rendimiento al bidiagonalizar matrices de variados tamaños cuando se utilizan distintas implementaciones variando la arquitectura. Determinando que algoritmo e implementación resulta más eficiente.
- Calcular la DVS utilizando el algoritmo identificado en el punto anterior, y finalmente implementarlo en el SRI desarrollado por el equipo.

V. DVS APLICADO A LA RECUPERACIÓN DE INFORMACIÓN

Se han ideado diferentes modelos basados en distintos paradigmas para representar tanto documentos como consultas en SRI y comparar la similitud de esas representaciones [3]. Entre estos se encuentran el modelo booleano, el modelo vectorial y el modelo probabilístico, denominados clásicos. El trabajo de investigación en curso cuya etapa inicial se presenta en este artículo se enmarca en una variante del método de RI vectorial, la LSI [2].

Con la LSI se define un espacio semántico donde los términos y los documentos altamente relacionados son colocados unos cerca de otros, reflejando los patrones de asociación entre los datos más importantes e ignorando los menos importantes, es decir los que tienen menor influencia al momento de la recuperación.

La técnica estadística particular que se aplica es la DVS de una matriz [2], [4]. Esta es una técnica ampliamente usada para descomponer una matriz en varias matrices que exhiben las propiedades más importantes de la matriz original. Así, una matriz A de tamaño t × d descompuesta con DVS (ver Fig. 1) produce tres matrices de la forma:

$$\mathbf{A = T_0 S_0 D_0}$$

[1] http://www.netlib.org/lapack/

ISSN: 2525-1333. Vol.:4-Nro.2 (Diciembre-2019)

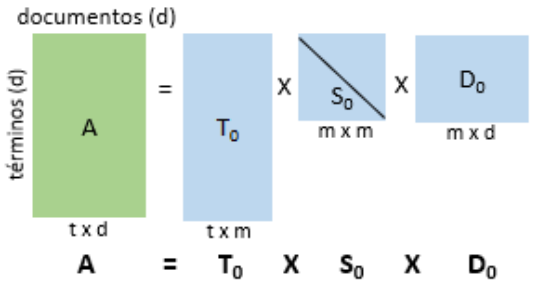

Fig. 1. Reducción de las dimensiones en DVS. Fuente [2]

T_0 y D_0 tienen columnas ortonormales (ortogonales y de tamaño uno) y son las matrices izquierda y derecha, respectivamente, de vectores singulares y S_0 es una matriz diagonal compuesta de los valores singulares de A.

La importancia de obtener modelos de orden reducido tiene que ver con que simplifican la comprensión del sistema, reducen el coste computacional en los problemas de simulación, implican menor esfuerzo computacional en el diseño de controladores numéricamente más eficientes y se obtienen leyes de control más simples [5].

Por ello es necesario buscar modelos matemáticos más simples que aproximen al máximo el comportamiento del sistema original. Este modelo, que poseerá menor número de estados que el sistema original, se denomina modelo reducido o modelo de orden reducido y al procedimiento utilizado para conseguirlo reducción de modelo.

Existen dos tipos principales de algoritmos utilizados en el cálculo computacional de la DVS de una matriz real: el método unilateral de Jacobi y algoritmos basados en bidiagonalización [6]. Este trabajo se enfoca en los algoritmos basados en bidiagonalización, los cuales aplican transformaciones ortogonales con el fin de obtener una forma bidiagonal para luego conseguir la DVS de la matriz bidiagonal.

VI. ALGORITMOS PARA BIDIAGONALIZACIÓN

La reducción bidiagonal de una matriz densa general se usa muy frecuentemente como un paso previo para calcular la DVS [7], [8].

A partir de la revisión en la literatura se descubrió que existen distintos métodos para la bidiagonalización de una matriz, los más tradicionales utilizan las transformaciones de Householder por la izquierda y por la derecha de la matriz [6], [9], [10]. Estudios realizados demuestran que estos presentan dos desventajas: cuando las matrices son de grandes dimensiones requieren tiempos de computación elevados y además repercuten negativamente en los costos de comunicación de una implementación paralela del algoritmo en sistemas de memoria distribuida [11], [12]. De hecho, según Ltaief [8], el número total de operaciones para dicho algoritmo sea $8/3\ n^3$, siendo n previsible de varios miles.

Con el énfasis puesto en dar una solución a estos problemas se han realizado diversos trabajos, entre estos se encuentran la propuesta de Ralha [13], mejorada más adelante por Barlow [14], orientada a conseguir un método más sencillo de paralelizar que los métodos tradicionales. En esta propuesta la bidiagonalización es unilateral, las transformaciones de Householder son aplicadas solamente por el lado derecho de la matriz.

Posteriormente Da Silva Sanches de Campos [12] presenta una mejora al método de Barlow con el objetivo de reducir el número de comunicaciones necesarias para una implementación paralela destinada a sistemas de memoria distribuida.

El método de bidiagonalización suele ser altamente paralelizable debido a las operaciones que utiliza en el proceso [15]. Vale mencionar que la correcta ejecución de algoritmos paralelos depende fuertemente de que los tamaños de las matrices se adapten a las capacidades de la máquina donde estos se ejecutan, por lo que, en matrices

de alta dimensionalidad, surgen problemas como el espacio en la memoria, la correctitud del algoritmo y el incremento en los tiempos de ejecución.

Con lo anterior presente, se han realizado numerosos trabajos que incluyen estudios comparativos en cuanto al rendimiento al bidiagonalizar matrices de variados tamaños cuando se utilizan distintas implementaciones variando la arquitectura. Entre estos, se han contrastado implementaciones secuenciales y paralelas sobre una arquitectura homogénea basada en CPU [12]; se han experimentado algoritmos en mosaico con distinta cantidad de nodos multinúcleo de un sistema de memoria compartida distribuida en paralelo [8], [16]. Otros han buscado aprovechar la capacidad que ofrecen las Unidades de Procesamiento Gráfico (GPU) y experimentaron su uso aplicando algoritmos en arquitecturas tanto homogéneas [17], [18] como también heterogéneas en las que se combinan el uso de CPU con GPU [19].

En este proyecto se decide poner especial interés en los algoritmos alternativos de bidiagonalización propuestos por Ralha y Barlow [13], [14], dado que están pensados para soportar el paralelismo, ya que la proyección a futuro es realizar la paralelización de los mismos a partir de una arquitectura basada en GPU.

A modo preliminar se decidió construir un algoritmo genérico en el lenguaje C# para el cálculo de la bidiagonalización basado en las transformaciones de Householder [8], el cual servirá de base tanto para comprender el proceso en sí mismo, como también, para tomarlo como referencia en el diseño e implementación de los otros dos algoritmos mencionados antes. El algoritmo 1 toma como entrada una matriz densa A y da como salida la descomposición bidiagonal superior. Los reflectores u_j y v_j pueden almacenarse en las partes inferior y superior de A, respectivamente. La mayor parte del cálculo se encuentra en la línea 5 y en la línea 10 en la que los reflectores se aplican a la matriz A desde la izquierda y luego desde la derecha, respectivamente. Se necesitan 4 flops para llevar a cero un elemento de la matriz, lo que hace que el número total de operaciones para dicho algoritmo sea 8/3 n^3.

Algoritmo 1 Reducción Bidiagonal via Reflectores de Householder

1: **for** $j = 1$ to n **do**
2: $x = A_{j:n,j}$
3: $u_j = sign(x_1)\ ||x||_2\ e_1 + x$
4: $u_j = u_j\ /\ ||u_j||_2$
5: $A_{j:n,j:n} = A_{j:n,j:n} - 2\ u_j\ (u_j^*\ A_{j:n,j:n})$
6: **if** $j < n$ **then**
7: $x = A_{j,j+1:n}$
8: $v_j = sign(x_1)\ ||x||_2\ e_1 + x$
9: $v_j = v_j\ /\ ||v_j||_2$
10: $A_{j:n,j+1:n} = A_{j:n,j+1:n} - 2\ (A_{j:n,j+1:n}\ v_j)\ v_j^*$
11: **end if**
12: **end for**

VI. CONCLUSIONES

Parte del trabajo de este proyecto de investigación tiene que ver con optimizar el proceso de bidiagonalización implementando para ello un algoritmo en una arquitectura híbrida basada en GPU. A modo preliminar, con el objetivo de lograr una mayor comprensión del proceso, se ha desarrollado un algoritmo en el lenguaje C# para la reducción de una matriz a su forma bidiagonal vía reflectores de Householder. El principal aporte de esta primera etapa tiene que ver con que dicho algoritmo ofrece evidencia de las funciones internas del proceso de bidiagonalización, algo que resulta engorroso de comprender cuando se utilizan rutinas de la biblioteca LAPACK. Vale aclarar que el algoritmo solo se ha realizado a modo demostrativo, aunque los resultados coinciden con los obtenidos al usar la biblioteca LAPACK, aun debe ser optimizado para futuras implementaciones en arquitecturas paralelas. Sin embargo, este ha sido desarrollado con la suficiente

flexibilidad para ser adaptado a futuro al momento de evaluar y comparar otras alternativas.

VII. REFERENCIAS Y BIBLIOGRAFÍA

A. Referencias bibliográficas:

[1] M. Mamani Roque. "Descomposición en Valores Singulares y Análisis Semántico Latente". *Tesis de Maestría*. Universidad Politécnica de Valencia, España, 2018.

[2] S. Deerwester, S. Dumais, G. Furnas, T. Landauer & R. Harshman. "Indexing by latent semantic analysis". *Journal of the American Society for Information Science*. 41(6):391–407. 1990.

[3] Tolosa G. & Bordignon, F. "Introducción a la Recuperación de Información: Conceptos, modelos y algoritmos básicos". Universidad Nacional de Luján, Argentina, 2008. Recuperado el 01/08/2019 de: http://eprints.rclis.org/12243/1/Introduccion-RI-v9f.pdf

[4] Berry, M., Dumais, S. & O'Brien, G. "Using Linear Algebra For Intelligent Information Retrieval". *Society for Industrial and Applied Mathematics*, Review 37(4): 573-595. Philadelphia, USA, 1995.

[5] L. Fortuna, G. Nunnari & A. Gallo. "Model order reduction techniques with applications in electrical engineering". *Springer-Verlag*, 1992.

[6] J. Demmel, M. Gu, S. Eisenstat, et al. "Computing the Singular Value Descomposition with Hight Relative Accuracy". *Linear Algebra and its Application,* 299, 21-80, 1999.

[7] Golub, G. & Van Loan, C. *Matrix Computation*. John Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Third Edition, 1996.

[8] Ltaief, H., Kurzak, J. & Dongarra, J. "Parallel Two-Sided Matrix Reduction to Band Bidiagonal Form on Multicore Architectures". *IEEE Transactions on Parallel and Distributed Systems*, 21(4): 417 – 423, 2010.

[9] G. Golub & C. Reinsch. "Singular Value Decomposition and Least Squares Solutions", *Handbook Series Linear Algebra*, 14: 403-420, 1970.

[10] T. Chan. "An Improved Algorithm for Computing the Singular Value Decomposition". *ACM Transactions on Mathematical Software*, 8(1): 72-83, 1982.

[11] Sangwine, S. & Le Bihan, N. "Quaternion Singular Value Decomposition based on Bidiagonalization to a Real Matrix using Quaternion Householder Transformations" *Applied Mathematics and Computation*, *ELSEVIER*, 182(1): 727-738, 2006.

[12] Da Silva Sanches de Campos, C. "Algoritmos de Altas Prestaciones para el Cálculo de la Descomposición en Valores Singulares y su Aplicación a la Reducción de Modelos de Sistemas Lineales de Control". Tesis Doctoral. Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, España, 2014.

[13] Ralha, R. "One-sided reduction to bidiagonal form". *Linear Algebra and Its Applications, ELSEVIER*, 358(1-3): 219-238, 2003.

[14] Barlow, J., Bosner, N., Drmač, Z. "A new stable bidiagonal reduction algorithm". *Linear Algebra and Its Applications, ELSEVIER*, 397: 35-84, 2005.

[15] Guerrero López, D. "Algoritmos Paralelos para la Reducción de Sistemas Lineales de Control Estables". Tesis doctoral. Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, España, 2015.

[16] Faverge M., Langou, J., Robert, Y. & Dongarra, J. "Bidiagonalization and R-Bidiagonalization: Parallel Tiled Algorithms, Critical Paths and Distributed-

Memory Implementation". *IEEE Transactions on Parallel and Distributed Processing Symposium*, 668 - 677, 2017.

[17] Lahabar, S. & Narayanan, P. "Singular Value Decomposition on GPU using CUDA". *IEEE International Symposium on Parallel & Distributed Processing*, 1-10, 2009.

[18] Dong, T., Haidar, A., Tomov, S. & Dongarra, J. "Optimizing the SVD Bidiagonalization Process for a Batch of Small Matrices". *Linear Algebra and Its Applications, ELSEVIER*, 108: 1008-1018, 2017.

[19] Hernández Cortés, J. "Implementación paralela y heterogénea de la transformación de Householder y sus aplicaciones".Tesis de Maestría. Departamento de Computación, Unidad Zacatenco, México, 2017.

B. Bibliografía:

A. Howard, C. Rorres. *Elementary Linear Algebra*. Wiley. USA, 11th edition, 2017.

Recibido: 2019-12-27
Aprobado: 2020-01-23
Hipervínculo Permanente: https://reddi.unlam.edu.ar
Datos de edición: Vol. 4 - Nro. 2 -Art. 5
Fecha de edición: Formato: 2020-01-31

Hacia la Optimización de un Sistema de Recuperación de Información

Osvaldo Spositto[1], Viviana Ledesma[1], Gastón Procopio[1], Hugo Ryckeboer[1]
[1]Departamento de Ingeniería e Investigaciones Tecnológicas,
Universidad Nacional de La Matanza
{spositto, vledesma, gprocopio, hugor}@unlam.edu.ar

RESUMEN

Con el desarrollo de los repositorios digitales cada vez ha cobrado mayor interés el estudio de los Sistemas de Recuperación de Información. El volumen de la información contenida en dichos repositorios crece de forma exponencial con lo cual búsqueda de los documentos que respondan a la necesidad de los usuarios se torna una tarea difícil. En este contexto este grupo de investigación ha estado trabajando por más de ocho años en la construcción de sus propios motores de búsqueda y recuperación orientados a corpus estáticos. Una vez construidos dichos motores las líneas de investigación se han orientado a distintos enfoques que pretenden acelerar los mismos tanto en la búsqueda como en los preprocesos. En particular, en esta etapa se tiene por objetivo la investigación, desarrollo e implementación de algoritmos paralelos, principalmente para resolver el proceso de la de Descomposición en Valores Singulares en arquitecturas basadas en Unidades de Procesamiento Gráfico y su comparación con clústeres de multicores, así como el empleo de soluciones híbridas que combinen ambos enfoques.

Palabras clave: Sistemas de Recuperación de Información, Descomposición en Valores Singulares, Bidiagonalización, Indexación Semántica Latente.

CONTEXTO

La línea de investigación que se presenta se encuentra inmersa en el proyecto de investigación C225 "*Resolución Eficiente de la Descomposición en Valores Singulares en una Arquitectura Híbrida y su Posterior Inserción en un Sistema de Recuperación de Información*" llevado a cabo en el marco del programa PROINCE de la Universidad Nacional de La Matanza (UNLaM). El mismo se desarrolla en el Polo Tecnológico dependiente del Departamento de Ingeniería e Investigaciones Tecnológicas de la UNLaM.

Este trabajo continúa con la línea de investigación de los proyectos PROINCE C151, C177 y C205 cuya temática se orientó: primero al estudio del tema y posteriormente a la realización de un prototipo de un sistema de recuperación de la información, la optimización de la recuperación de documentos usando como técnica base el LSI (Lematización Semántica Latente), y el uso de Minería de Datos para acelerar la recuperación de documentos.

1. INTRODUCCIÓN

A partir de la década del 90 los avances tecnológicos de la informática dieron lugar a un incremento exponencial en la generación y almacenamiento de información que continúa hasta la actualidad. La enorme cantidad de información almacenada hace que su búsqueda y recuperación sea cada vez más dificultosa, lo cual impulsó el estudio de la Recuperación de la Información (RI) como disciplina. Existen dos tendencias principales en el desarrollo de Sistemas de RI (SRI) según el contexto y el ámbito de la fuente documental [1]: La RI vertical que se enfoca en la indexación de fuentes documentales específicas, por ejemplo, una biblioteca de Ciencias Jurídicas. La RI horizontal que se enfoca en fuentes documentales generales, por ejemplo, la Web. Esta línea de investigación se relaciona con el primer grupo dado que se trata la implementación y optimización de un SRI de fuentes documentales específicas y la evaluación de su rendimiento.

En la literatura existen diversas propuestas sobre la organización interna de un SRI [2, 3]. La Figura 1 muestra una representación simplificada de un SRI, tal como se visualiza, los procesos más importantes que intervienen

en dicho sistema son los siguientes:

Indexación – los documentos que alimentan el sistema se representan como objetos indexados.

Búsqueda – se analiza la consulta del usuario y se compara con los objetos indexados, de tal modo se pueden obtener los objetos recuperados que se le presentarán al usuario.

Ranking – se determina la relevancia de cada documento recuperado para dar solución a la consulta que haya ingresado el usuario, finalmente los documentos se ordenan en base a los valores obtenidos en este proceso.

Este trabajo está enmarcado dentro del proceso de indexación.

Figura 1. Sistema de RI.

Se han ideado diferentes modelos basados en distintos paradigmas para representar tanto documentos como consultas en SRI y comparar la similitud de esas representaciones [4]. Entre estos se encuentran el modelo booleano, el modelo vectorial y el modelo probabilístico, denominados clásicos. Esta línea de investigación se enmarca en una variante del método de RI vectorial, la Indexación Semántica Latente (ISL) [5].

La ISL es un método para la búsqueda de información en documentos a través de la indexación de términos [5]. Con dicho método se pretende resolver perturbaciones en la RI causados por problemas de sinonimia y polisemia o equivocidad del habla corriente. Para ejemplificar, si se desea buscar la palabra "estación", la cual tiene múltiples significados (polisemia) una búsqueda literal de la palabra produciría muchos resultados posibles (estación de tren, estación del año, etc.). Si lo que se desea buscar es "estación del año", podrían interesar resultados de palabras distintas, pero con un significado igual o similar, por ejemplo "temporada", "época" y así por el estilo (sinonimia). La ISL permite la búsqueda por conceptos o definiciones en contraposición a la búsqueda literal.

La aplicación de la ISL implica la utilización de algoritmos matemáticos especializados, que como resultado simulan el análisis que realizaría una persona. Una técnica ampliamente utilizada a tal fin es la Descomposición en Valores Singulares (DVS), luego la recuperación se realiza utilizando como punto de partida los valores singulares y vectores obtenidos a partir de la aplicación de dicha técnica [6].

La DVS [5,7] consiste en descomponer una matriz en varias matrices que exhiben las propiedades más importantes de la matriz original. Así, una matriz A de tamaño t × d descompuesta con DVS (ver Figura 2) produce tres matrices de la forma:

$$A = T_0 S_0 D_0$$

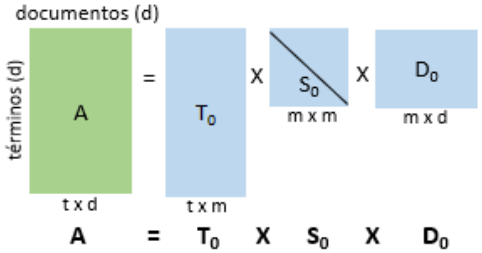

Figura 2. Reducción de dimensiones en DVS. Fuente: [5]

T_0 y D_0 tienen columnas ortonormales (ortogonales y de tamaño uno) y son las matrices izquierda y derecha, respectivamente, de vectores singulares y S_0 es una matriz diagonal compuesta de los valores singulares de A.

Disponer de modelos de orden reducido tiene como ventaja el simplificar la comprensión del sistema, reducir el coste computacional en los problemas de simulación, lo cual a su vez implica menor esfuerzo computacional en el diseño de controladores numéricamente más eficientes y se obtienen leyes de control más simples [8]. De ahí la necesidad de buscar modelos

matemáticos simplificados que aproximen al máximo el comportamiento del sistema original. El modelo resultante, que poseerá un número menor de estados que el sistema original, se denomina modelo reducido o modelo de orden reducido y al procedimiento utilizado para conseguirlo se lo conoce como reducción de modelo.

Existen dos tipos principales de algoritmos que se aplican al cálculo computacional de la DVS de una matriz real, el método unilateral de Jacobi y aquellos algoritmos que se basan en la bidiagonalización [6]. El número de operaciones para los distintos algoritmos se encuentra en el orden de On^3, las diversas propuestas y mejoras que han surgido buscan disminuir operaciones costosas en tiempo.

Mediante este proyecto se pretende optimizar la resolución de la DVS, en especial mediante implementar algoritmos para resolver la primera fase de este proceso, a través de la bidiagonalización. Los métodos más tradicionales de bidiagonalización utilizan las transformaciones de Householder por la izquierda y por la derecha de la matriz [9, 10]. Como desventaja, cuando las matrices son de grandes dimensiones requieren tiempos de computación elevados y además repercuten negativamente en los costos de comunicación de una implementación paralela del algoritmo en sistemas de memoria distribuida [11,12]. Así es que se han realizado diversos trabajos, entre estos se encuentran la propuesta de Ralha [13], mejorada más adelante por Barlow [14], orientada a conseguir un método más sencillo aplicando las transformaciones de Householder solamente por el lado derecho de la matriz. Posteriormente, Da Silva Sanches de Campos [12] presenta una mejora al método de Barlow con el objetivo de reducir el número de comunicaciones necesarias para una implementación paralela destinada a sistemas de memoria distribuida.

Para este proyecto se pondrá especial interés en los algoritmos alternativos de bidiagonalización propuestos por Ralha y Barlow [13,14], dado que están pensados para soportar el paralelismo. La decisión se sustenta en que el resultado final de esta etapa de la investigación se orienta hacia algoritmos que puedan ser implementados en plataformas paralelas y, en particular aprovechando la capacidad de las unidades de procesamiento gráfico (GPU, por sus siglas en inglés).

2. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

A lo largo de varios proyectos de investigación, se ha profundizado la investigación de la RI. La meta inicial del equipo fue construir íntegramente un prototipo de un sistema de organización de documentos para su posterior recuperación mediante un buscador. Habiendo concluido esa primera etapa se buscó optimizar el trabajo de RI lo cual implica dar atención a distintas líneas de investigación:

a) Mejorar la lematización disponible, en particular para este caso del idioma español;
b) Subdividir el corpus en forma inteligente de modo tal que sin gran pérdida de exhaustividad se pueda resolver la consulta examinando una o más partes de la subdivisión, excluyendo a muchas de ellas; y
c) Acelerar la velocidad de cómputo.

En particular, esta etapa de la investigación se relaciona a la línea de investigación mencionada en el ítem c. El objetivo principal de este trabajo apunta a diseñar, implementar y evaluar algoritmos secuenciales y paralelos para resolver eficientemente cada uno de los algoritmos utilizados en la DVS. En función a tal objetivo se han trazado los siguientes objetivos específicos:

a) Estudiar el problema matemático de la DVS y las variantes algorítmicas existentes para una mejor implantación en GPU.
b) Estudiar a bajo nivel las arquitecturas de los equipos CPU y GPU, sobre los cuales se realizarán los desarrollos, así como de las herramientas de software necesarias para su máximo aprovechamiento.
c) Estudiar las principales librerías disponibles que resuelven problemas relacionados con el álgebra lineal, especialmente aquellas que lo realicen en

arquitectura de cálculo paralelo, por ejemplo, CUDA, CUBLAS, OPENCL, etc.

d) Desdoblamiento de los algoritmos para resolverlos sobre una configuración híbrida.
e) Realizar un estudio comparativo en cuanto al rendimiento al bidiagonalizar matrices de variados tamaños cuando se utilizan distintas implementaciones variando la arquitectura. Determinando que algoritmo e implementación resulta más eficiente.
f) Calcular la DVS utilizando el algoritmo identificado en el punto anterior, y finalmente implementarlo en el SRI desarrollado por el equipo.

3. RESULTADOS OBTENIDOS/ESPERADOS

A continuación, se enumeran los resultados ya alcanzados:

i. Se construyeron dos prototipos de SRI, uno basado en lo que ahora se pueden llamar métodos clásicos y otro según el método LSI. Estos se diseñaron de forma modular de tal manera que permitan mejoras locales con su consiguiente experimentación. Los mismos se almacenaron con código abierto a fin de facilitar a nuevos grupos interesados en esta tecnología para su iniciación en el tema y la base sobre la cual efectuar experimentos propios.
ii. Se extendió la selección de documentos a corpus voluminosos a través de la utilización de DVS y clustering. La DVS proporciona una salida donde las diferencias de las distancias entre los documentos son muy cercanas, mientras que clustering utiliza esas distancias para poder agrupar los documentos, parte de este trabajo ha sido publicado en [15].
iii. Se estudió como línea alternativa fraccionar el corpus. Esto requiere dos algoritmos preparatorios, uno que particione el corpus utilizando una noción de vecindad o similitud y el entrenamiento de un algoritmo de clasificación que direccione la consulta hacia la parte más promisoria. Para ambos servicios se aplicaron técnicas de minería de datos y de la selección de la parte usando redes neuronales. Los resultados se han reflejado parcialmente en [16].
iv. Adicionalmente se desarrolló un generador de corpus, para ser utilizado en esta línea de investigación y pretendiendo, además, con esto, colaborar con la propagación de la Lingüística de Corpus como metodología para investigaciones en RI, el trabajo realizado se describe en [17].

En las próximas etapas de investigación, se profundizará el análisis de la posibilidad de optimizar la DVS, dando atención en particular a la primera fase, la bidiagonalización. Para ello se realizará un estudio comparativo en cuanto al rendimiento al bidiagonalizar matrices de variados tamaños cuando se utilizan distintos algoritmos y distintas implementaciones variando la arquitectura. De esta manera se intentará determinar qué algoritmo e implementación resulta más eficiente. La investigación se centrará en el desarrollo e implementación de algoritmos paralelos, principalmente el de DVS, en arquitecturas basadas en GPU y su comparación con clústeres de multicores, así como el empleo combinado de GPU y multicores. El alto grado de paralelismo de las GPU que sin lugar a duda disminuye el tiempo de cálculo, sufre una mengua en la misma a causa de la lentitud de sus comunicaciones transversales. De allí surge el interés del equipo en explorar si soluciones híbridas pudieran aportar una aceleración en los cómputos, delegando en cada parte, CPU o GPU, aquellas tareas en las cuales mejor se desempeñan.

Por otro lado, se llevará a cabo la puesta a prueba de dicha optimización en el SRI desarrollado por el equipo con el fin de comprobar el nivel de impacto alcanzado en la productividad del proceso. No es objetivo de este proyecto obtener una resolución en forma abstracta y genérica como para enriquecer las bibliotecas del cálculo matricial, sino resolverlo para ciertas arquitecturas concretas disponibles en la sede del proyecto. Esto marcaría el camino para que otros, con más recursos económicos y tamaño del equipo

humano, puedan extenderlo y parametrizarlo para que funcione en otras configuraciones.

4. FORMACIÓN DE RECURSOS HUMANOS

En el proyecto participan seis investigadores, uno de ellos en formación y dos son alumnos de grado. La línea de investigación presentada aquí es parte directa de la tesis "*Estudio comparativo de DBSCAN, KMEANS con redes neuronales en un Sistema de Recuperación de Información*", correspondiente a la Maestría en Informática que está desarrollando el Ing. Casuscelli Marcos en UNLaM.

Durante el último año la Ing. Viviana Ledesma presentó su tesis de maestría y su posterior defensa, la cual desarrolló en la UNLaM, y por su parte, el alumno Gastón Procopio, finalizó su carrera de Ingeniería en Informática con la aprobación del proyecto final de carrera.

Parte de los resultados de esta investigación son divulgados en la cátedra de Diseño de Sistemas que se dicta para la carrera de Ingeniería en Informática de la UNLaM. Se espera además que esta investigación contribuya a la formación de recursos humanos en RI y que el sistema desarrollado pueda servir de base para una transferencia de tecnología a las PYMEs de la región.

5. BIBLIOGRAFÍA

[1] Cleverdon, C.W. "Progress in documentation. Evaluation of information retrieval systems", Journal of Documentation, 26, 55-67, 1970.

[2] Kowalski, G. "Information Retrieval Systems: Theory and Implementation", 1st ed. Norwell, MA, USA: Kluwer Academic Publishers, 1997.

[3] Kuna, H., Rey, M., Martini, E., Solonezen, L. & Podkowa, L. "Desarrollo de un Sistema de Recuperación de Información para Publicaciones Científicas del Área de Ciencias de la Computación", Revista Latinoamericana de Ingeniería de Software, 2014. 2(2): 107-114.

[4] Tolosa G. & Bordignon, F. "Introducción a la Recuperación de Información: Conceptos, modelos y algoritmos básicos". Universidad Nacional de Luján, Argentina, 2008. Recuperado el 01/08/2019 de: http://eprints.rclis.org/12243/1/Introduccion-RI-v9f.pdf

[5] Deerwester, S., Dumais, S., Furnas, G., Landauer, T. & Harshman, R. "Indexing by latent semantic analysis". Journal of the American Society for Information Science (SIAM),1990. 41(6):391–407.

[6] Lahabar, S. & Narayanan, P. "Singular Value Decomposition on GPU using CUDA". IEEE International Symposium on Parallel & Distributed Processing, 2009. 1-10.

[7] Berry, M., Dumais, S. & O'Brien, G. "Using Linear Algebra For Intelligent Information Retrieval". Society for Industrial and Applied Mathematics, Review 37(4): 573-595. Philadelphia, USA, 1995.

[8] L. Fortuna, G. Nunnari & A. Gallo. "Model order reduction techniques with applications in electrical engineering". Springer-Verlag, 1992.

[9] J. Demmel, M. Gu, S. Eisenstat, et al. "Computing the Singular Value Descomposition with Hight Relative Accuracy". Linear Algebra and its Application, 299, 21-80, 1999.

[10] T. Chan. "An Improved Algorithm for Computing the Singular Value Decomposition". ACM Transactions on Mathematical Software, 8(1): 72-83, 1982.

[11] Sangwine, S. & Le Bihan, N. "Quaternion Singular Value Decomposition based on Bidiagonalization to a Real Matrix using Quaternion Householder Transformations" Applied Mathematics and Computation, ELSEVIER, 182(1): 727-738, 2006.

[12] Da Silva Sanches de Campos, C. "Algoritmos de Altas Prestaciones para el Cálculo de la Descomposición en Valores Singulares y su Aplicación a la Reducción de Modelos de Sistemas Lineales de Control". Tesis Doctoral. Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, España, 2014.

[13] Ralha, R. "One-sided reduction to bidiagonal form". Linear Algebra and Its Applications, ELSEVIER, 358(1-3): 219-238, 2003.

[14] Barlow, J., Bosner, N., Drmač, Z. "A new stable bidiagonal reduction algorithm". Linear Algebra and Its Applications, ELSEVIER, 397: 35-84, 2005.

[15] Spositto, O., Procopio, G., Quintana, F.& Ryckeboer H. "Una paralelización del método de Householder", CACIC 2016, pp. 1291-1300. Universidad Nacional de San Luis San Luis, 2016.

[16] Spositto, O., Casuscelli, M., Bossero, J., Matteo, L., Ryckeboer, H. "Aceleración en la Recuperación de Información utilizando Algoritmos de Minería de Datos de R". CACIC 2018, pp.491-500.Universidad Nacional del Centro, Tandil, 2018.

[17] Spositto, O., Procopio, G. Bossero, J. "Método para la Construcción de un Corpus Periodístico mediante Expresiones Regulares". CONAIISI 2018, pp. 491-500. Universidad CAECE, Mar del Plata, 2018.

Comparación de un Algoritmo de Bidiagonalización para su Utilización en la Recuperación de Información

Osvaldo Spositto[1], Viviana Ledesma[1], Gastón Procopio[1], Hugo Ryckeboer[1], Victoria Saizar[1] y Alexis Vainberg[1]

[1] Departamento de Ingeniería e Investigaciones Tecnológicas,
Universidad Nacional de La Matanza, Buenos Aires, Argentina.
{spositto; vledesma, gprocopio, hugor, vsaizar, avainberg
}@unlam.edu.ar

Abstract. Este artículo presenta parte del trabajo realizado en el marco de una investigación que pretende optimizar un Sistema de Recuperación de Información, mediante la implementación y evaluación de distintos algoritmos secuenciales y paralelos para resolver eficientemente la Descomposición en Valores Singulares. Tal proceso comienza con llevar la matriz inicial a la forma bidiagonal, lo que puede consumir más del 70% del tiempo total del proceso. Por ello, como trabajo preliminar se han estudiado distintos métodos de bidiagonalización. Este trabajo se relaciona al desarrollo e implementación de un algoritmo de bidiagonalización alternativo para comparar posteriormente su comportamiento en distintas arquitecturas, en particular, las basadas en unidades de procesamiento gráfico, monoprocesadores y multiprocesadores. La experiencia de este estudio concreto ha permitido un análisis de rendimiento al ejecutar el algoritmo en cada implementación, cuando se varía el tamaño de las matrices, identificando problemas mínimos en GPU en cuanto a diferencias en la precisión de datos.

Keywords: Descomposición de Valores Singulares, Bidiagonalización, Sistema de Recuperación de Información.

1 Introducción

La Indexación Semántica Latente (ISL) es un método para la búsqueda de información en documentos a través de la indexación de términos [1], lo cual involucra la aplicación de algoritmos matemáticos especializados a fin de simular el análisis que realizaría una persona. Una técnica ampliamente utilizada a tal fin es la Descomposición en Valores Singulares (DVS), luego la recuperación se realiza a partir de los valores y vectores singulares obtenidos al aplicar dicha técnica [2].

Este trabajo se realiza en el contexto de un proyecto de investigación que tiene por objetivo optimizar la resolución de la DVS, para su posterior inserción en un Sistema de Recuperación de Información (SRI) desarrollado por el mismo equipo. En particular, la mejora se enfoca en implementar algoritmos que permitan resolver la primera fase del proceso de la DVS, conocido como bidiagonalización [3]. Se ha

comprobado que esta fase es la que más tiempo insume, estudios realizados muestran que puede consumir más del 70% o 90% del tiempo total para obtener todos los vectores singulares o solo los valores singulares, respectivamente [4]. Esto hace que el método de bidiagonalización, con un alto nivel de paralelismo de sus operaciones, sea un excelente candidato para la utilización de unidades de procesamiento gráfico (GPU, por sus siglas en inglés) ya que estas brindan la potencia de procesamiento requerida. Son varios los autores que han presentado trabajos en los que proponen la utilización de GPU para la bidiagonalización [5], [6].

En principio se han evaluado distintos métodos de bidiagonalización y, a modo inicial, se implementó un algoritmo genérico, secuencial, que sirvió para evidenciar el funcionamiento interno del proceso. Posteriormente, se ha orientado el estudio hacia algoritmos que puedan ser implementados en plataformas paralelas.

Luego de estudiar distintas variantes, se ha decidido adaptar y desarrollar un algoritmo alternativo propuesto por Barlow, Bosner y Drmač [7] (en adelante, Barlow), para evaluar su implementación en tres arquitecturas diferentes, basadas en CPU monoprocesador, multiprocesador y en GPU. Vale aclarar que se han encontrado trabajos en los que el algoritmo en estudio es probado en MatLab, en CPU secuencial y paralelo, pero nunca ha sido probado en GPU. Para la implementación se utilizó el framework de CUDA con el fin de comparar los tiempos de respuesta resultantes cuando se aplica el algoritmo en matrices de distintos tamaños.

El resto del artículo se organiza de la siguiente manera: la Sección 2 repasa los algoritmos de bidiagonalización en el contexto de los métodos utilizados para la recuperación de información; la Sección 3 describe los algoritmos implementados para este trabajo; la Sección 4 muestra los resultados experimentales obtenidos; y finalmente, la Sección 5 presenta las principales conclusiones e ideas para avanzar en esta investigación.

2 Métodos para la Recuperación de Información

Un SRI necesita componerse, por una parte, de un formalismo que permita representar documentos y consultas y, por otra parte, de una medida de similitud entre un documento y una consulta. En la actualidad conviven una variedad de modelos basados en distintos paradigmas para representar tanto documentos como consultas en SRI y comparar la semejanza de tales representaciones [8]. Entre estos, se destacan los modelos clásicos: el modelo booleano, el modelo vectorial y, el modelo probabilístico. En el modelo vectorial se seleccionan las palabras útiles, que por lo general son todos los términos del documento a excepción de las palabras semánticamente vacías, este proceso se enriquece utilizando técnicas de lematización y etiquetado [9]. El trabajo presentado en este artículo está circunscripto en una variante del método de recuperación vectorial, la ISL.

El método de ISL permite la búsqueda de información en documentos mediante la indexación de sus términos [1]. Involucra la definición de un espacio semántico donde los términos y los documentos altamente relacionados son colocados unos cerca de otros, reflejando los patrones de asociación entre los datos más importantes e

ignorando los menos importantes, es decir los que tienen menor influencia al momento de la recuperación.

La aplicación de la ISL, como se dijo antes, implica la utilización de algoritmos matemáticos especializados, que como resultado simulan el análisis que realizaría una persona. Por otro lado, con el método de ISL se pretende resolver dificultades durante la recuperación causadas por problemas de sinonimia y polisemia (o equivocidad del habla corriente). Por ejemplo, si la búsqueda se realiza a partir de la palabra "estación", la cual tiene múltiples significados (polisemia) una búsqueda literal de la palabra produciría muchos resultados posibles (estación de tren, estación del año, etc.). Si lo que se desea buscar es "estación del año", resultaría de interés que los resultados incluyan palabras distintas, pero con un significado igual o parecido, por ejemplo "temporada", "época" y así por el estilo (sinonimia). Por lo tanto, aplicando la ISL es posible buscar por conceptos o definiciones en contraste a lo que sería una búsqueda literal. Para ello, un primer recurso es trabajar con lexemas y no con palabras, ya que palabras derivadas de una misma raíz comparten buena parte de la carga semántica.

En general, al indexar los términos de los documentos, la matriz de documentos resultante se vuelve muy grande, por tal razón, a fin de acelerar el proceso de recuperación de información, suelen aplicarse técnicas de reducción de la dimensionalidad con el fin de transformar dicha matriz en una de menores dimensiones, pero capaz de reflejar las características de la matriz original al momento de llevar adelante las búsquedas. Para tal propósito se aplica la DVS, una técnica de factorización de matrices que permite descomponer una matriz en varias matrices que presentan las propiedades más significativas de la matriz original [1], [10], [11]. Así, una matriz A de tamaño t × d descompuesta con DVS produce tres matrices, tal como se puede observar en la figura 1.

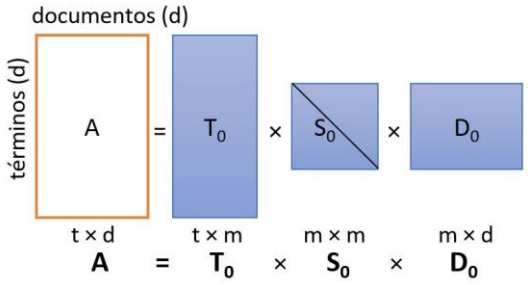

Fig. 1. Reducción de dimensiones en DVS. Fuente: [1].

Las columnas de T_0 y D_0 son ortonormales (ortogonales y de tamaño uno) y son las matrices izquierda y derecha respectivamente, de vectores singulares y, S_0 es una matriz diagonal compuesta de los valores singulares de A. El triple producto indicado da una matriz de t × d de rango m. De todas las matrices de t × d de rango m que aproximen a A, la de menor error, es decir distancia, es una que comparte los mayores m autovalores de A, obtenidos en una descomposición DVS y anula los restantes, comparte sus autovectores. Habiendo autovalores nulos sus correspondientes

componentes en los autovalores no tienen influencia y por lo tanto son recortados a tamaño m.

La ventaja de utilizar estos modelos de orden reducido es que simplifica la comprensión del sistema, reduce el coste computacional en los problemas de simulación, lo cual a su vez implica menor esfuerzo computacional en el diseño de controladores numéricamente más eficientes y se obtienen leyes de control más simples [12]. Esto justifica la importancia y necesidad de buscar modelos matemáticos simplificados que aproximen al máximo el comportamiento del sistema original. El modelo resultante, que tendrá un número menor de estados que el sistema original, se denomina modelo reducido o modelo de orden reducido, mientras que se conoce como reducción del modelo al procedimiento utilizado para conseguirlo.

Existen dos tipos principales de algoritmos que se aplican al cálculo computacional de la DVS de una matriz real, el método unilateral de Jacobi y aquellos que se basan en la bidiagonalización [10]. El número de operaciones para los distintos algoritmos se encuentra en el orden de $O(n^3)$, las diversas propuestas y mejoras que han surgido buscan disminuir operaciones costosas en tiempo. El trabajo de este equipo de investigación se enfoca en los algoritmos basados en bidiagonalización, los cuales aplican transformaciones ortogonales con el fin de obtener una forma bidiagonal para luego conseguir la DVS de la matriz bidiagonal.

2.1 Algoritmos Aplicados para la Bidiagonalización

Tal como se explicó anteriormente, la reducción bidiagonal de una matriz densa general se usa muy frecuentemente como un paso preliminar para el cálculo de la DVS [13]. A partir de la revisión en la literatura se descubrió que existen distintos métodos para la bidiagonalización de una matriz, los enfoques más tradicionales utilizan las transformaciones de Householder por la izquierda y por la derecha de la matriz [10], [14], [15]. Algunos estudios demuestran que dichos métodos presentan dos desventajas: cuando las matrices son de grandes dimensiones requieren tiempos de computación elevados y además repercuten negativamente en los costos de comunicación de una implementación paralela del algoritmo en sistemas de memoria distribuida [16], [17]. De hecho, según Ltaief [13], el número total de operaciones para dicho algoritmo sea $8/3(n^3)$, pudiendo ser n previsible de varios miles.

Pretendiendo dar una solución a tales problemas han surgido diversos trabajos, entre estos se encuentran la propuesta de Ralha [18], mejorada más adelante por Barlow [7], orientada a conseguir un método más sencillo de paralelizar que los métodos tradicionales. En esta propuesta la bidiagonalización es unilateral, es decir, las transformaciones de Householder son aplicadas solamente por el lado derecho de la matriz. Posteriormente, Da Silva Sanches de Campos [17] presenta una mejora al método de Barlow con el objetivo de reducir el número de comunicaciones necesarias para una implementación paralela destinada a sistemas de memoria distribuida.

Las operaciones utilizadas en el proceso hacen que el método de bidiagonalización sea altamente paralelizable [19]. De más está decir, que la correcta ejecución de algoritmos paralelos depende fuertemente de que los tamaños de las matrices se adapten a las capacidades de la máquina donde estos se ejecutan, por lo que, en

matrices de alta dimensionalidad, aparecen problemas como el espacio en la memoria, la correctitud del algoritmo y el incremento en los tiempos de ejecución.

Con lo anterior presente, se han realizado numerosos trabajos que incluyen estudios comparativos en cuanto al rendimiento al bidiagonalizar matrices de distintos tamaños cuando se utilizan distintas implementaciones variando la arquitectura. Entre estos, se han contrastado implementaciones secuenciales y paralelas sobre una arquitectura homogénea basada en CPU [17], se han experimentado algoritmos en mosaico con distinta cantidad de nodos multinúcleo de un sistema de memoria compartida distribuida en paralelo [4], [20]. Otros han buscado aprovechar la capacidad que ofrecen las GPU y experimentaron su uso aplicando algoritmos en arquitecturas tanto homogéneas [5], [3] como también heterogéneas en las que se combinan el uso de CPU con GPU [21].

En este trabajo se decide poner especial interés en uno de los algoritmos alternativos de bidiagonalización, el propuesto por Barlow en [7], dado que está pensado para soportar el paralelismo, lo cual está en consonancia con el objetivo de la investigación en curso, lograr una implementación a partir de una arquitectura basada en GPU.

3 Implementación de Algoritmos de Bidiagonalización

En principio se evaluaron distintos métodos para el cálculo de la bidiagonalización y, a modo inicial, se implementó un algoritmo genérico, secuencial, que sirvió para evidenciar el funcionamiento interno del proceso. Dicho algoritmo, basado en las transformaciones de Householder [4], expresado a continuación en la figura 2 como algoritmo, ha sido desarrollado en el lenguaje C#, parte de este trabajo ha sido presentado en [22]. La idea era que este, aunque secuencial, sirviera de base tanto para comprender el proceso en sí mismo, como también, para tomarlo como referencia en el diseño e implementación del algoritmo de Barlow.

Algoritmo 1: Reducción Bidiagonal vía Reflectores de Householder

```
1  for j = 1 to n do
2      x = A_{j:n,j}
3      u_j=sign(x_1) ||x||_2 e_1+ x
4      u_j= u_j/ ||u_j||_2
5      A_{j:n,j:n} =A_{j:n,j:n} - 2 u_j (u_j^* A_{j:n,j:n} )
6      if j < n then
7          x=A_{j,j+1:n}
8          v_j= sign(x_1) ||x||_2 e_1+ x
9          v_j= v_j / ||v_j||_2
10         A_{j:n,j+1:n} =A_{j:n,j+1:n} - 2 (A_{j:n,j+1:n} v_j) v_j^*
```

Fig. 2. Algoritmo de bidiagonalización basado en Householder. Fuente: [4].

Aunque existía la posibilidad de utilizar la biblioteca LAPACK[1], disponer del código en C# ofrece como ventaja, por una parte, permitir la comprensión de cada etapa interna del proceso, y por otra sienta las bases para que este código posteriormente pueda ser adaptado a diferentes algoritmos de bidiagonalización, e implementarlos en otras arquitecturas paralelas, en particular, aquellas basadas en GPU, a fin de analizar su eficiencia.

El algoritmo de Barlow, objeto de este estudio, y explicado en detalle en [7], consiste en un método para la bidiagonalización de matrices densas en el que las transformaciones de Householder se aplican únicamente por el lado derecho de la matriz. Con esto es posible definir todas las operaciones en términos de las columnas de la matriz a transformar, lo cual permite el desarrollo de implementaciones paralelas de un modo más simplificado en comparación con los métodos tradicionales, por otra parte, se logra reducir las comunicaciones que se necesitan.

El método de Barlow se puede así expresar, en forma de algoritmo, como se muestra en la figura 3, de la siguiente manera:

Algoritmo 2: BarlowBidiagonalización (A, α, β, Q)

1 **for** $r = 1, 2, ..., n-2$ **do**
2 $\quad \alpha_r = ||A(:,r)||_2$
3 $\quad q_r = \dfrac{A(:,r)}{\alpha_r}$
4 $\quad x_r = \mathrm{A}\ (:, r+1:n)^t\ q_r$
5 $\quad H_r$ tal que $H_r^t\ x_r = \beta_r\ e_1$
6 $\quad \mathrm{A}(:, \mathrm{r}+1:\mathrm{n}) = \mathrm{A}\ (:, \mathrm{r}+1:\mathrm{n})\ H_r$
7 $\quad \mathrm{A}(:, \mathrm{r}+1) = \mathrm{A}\ (:, \mathrm{r}+1) - \beta_r\ q_r$
8 **end**
9 $\alpha_{n-1} = ||A(:,n-1)||_2$
10 $q_{n-1} = \dfrac{A(:,n-1)}{\alpha_{n-1}}$
11 $\beta_{n-1} = q_{n-1}^t\ \mathrm{A}\ (:, \mathrm{n})$
12 $\mathrm{A}(:, \mathrm{n}) = \mathrm{A}\ (:, \mathrm{n}) - \beta_{n-1} q_{n-1}$
13 $\alpha_n = ||A(:,n)||_2$
14 $q_n = \dfrac{A(:,n)}{\alpha_n}$

Fig. 3. Algoritmo de bidiagonalización unilateral de Barlow. Fuente: [17].

Como se puede observar, hay un ciclo principal en el que se trabaja la matriz principal por columnas, va desde la columna *0* hasta la antepenúltima columna *(n-2)*. Además de la matriz inicial, hay 3 variables principales que se utilizan a lo largo de todo el algoritmo:

- α: es un vector de n elementos que contiene los valores de la diagonal principal.
- β: es un vector de n-1 elementos que contiene los valores de la diagonal superior.

[1] http://www.netlib.org/lapack/

- q: es una matriz con idénticas dimensiones que la matriz principal, es una matriz de trabajo interno.

En cada iteración se va completando: una posición en el vector α de elementos de la diagonal principal; una posición en el vector β de elementos de la diagonal superior, esto se representa en las líneas 5 y 6 del algoritmo en las que se aplican las reflexiones de Householder; una columna de la matriz q; y además, se modifica la matriz inicial que luego se lee en las iteraciones subsiguientes. Al terminar el ciclo se completan las posiciones restantes de α, β y las columnas que quedan de la matriz q.

Este algoritmo ha sido paralelizado y desarrollado para ser implementado sobre las tres arquitecturas mencionadas previamente, monoprocesador, multiprocesador y GPU, con el fin de comparar el rendimiento en cada una de estas. A continuación se resumen algunos resultados obtenidos.

4 Resultados Experimentales

Las implementaciones han sido desarrolladas utilizando el lenguaje C#, en conjunto con el framework CUDA, versión 6.5.

Las características del equipo utilizado para las pruebas de este experimento son:
- CPU: AMD Ryzen 5 2600 6 nucleos 12 threads a 3.6 GHz
- Memoria: 2 x 8GB DDR4 Crucial Ballistix 2400 Mhz
- GPU: NVIDIA GEFORCE GTX 1050 2GB

En la tabla 1 se presentan los tiempos de ejecución en milisegundos de cada una de las implementaciones realizadas para este estudio. Para las pruebas se utilizaron matrices cuadradas de distintas dimensiones, y tomando como fundamento las conclusiones obtenidas por Da Silva Sanches de Campos en [17], estas contienen valores aleatorios, dado que estos no tienen incidencia en los resultados esperados.

Tabla 1. Tiempos de ejecución del algoritmo en milisegundos para cada arquitectura

Dimensión Matriz	GPU	CPU Monoprocesador	CPU Multiprocesador
10x10	25	1	31
50x50	33	4	47
100x100	50	18	65
500x500	394	2005	1060
1000x1000	1691	16309	6008
2000x2000	10730	155418	34763

De los tiempos obtenidos, como resultado de las pruebas, se observa que cuando las matrices son de menor dimensión es conveniente ejecutar este tipo de algoritmos en CPU monoprocesador. Cuando la matriz comienza a superar las dimensiones, aproximadamente a partir de 200*200 o 300*300, la GPU mejora notoriamente el tiempo de ejecución con respecto a CPU monoprocesador y CPU multiprocesador. Esto se pone en evidencia, por ejemplo, observando los tiempos insumidos para la

matriz de tamaño 2000*2000, donde la GPU logró reducir los tiempos de ejecución en un 93% y 69%, con relación a CPU monoprocesador y multiprocesador respectivamente. En cuanto a la CPU multiprocesador, se puede observar que es constante el tiempo de resolución y este se incrementa lentamente hasta las dimensiones aproximadas entre 200*200 y 300*300.

Debe considerarse que en el proceso monoprocesador se realiza el proceso en serie, tomándose en un principio una columna, se aplica las operaciones matemáticas correspondientes y luego se modifica la matriz general en base al resultado de la columna resultante de dichas operaciones. En cambio, en el proceso multiprocesador las operaciones son distribuidas entre los núcleos disponibles, de esta manera cada thread resuelve una porción de las operaciones correspondientes acelerando de este modo la resolución del algoritmo.

Se ha recurrido a la presentación de un gráfico, que se muestra en la figura 4, en donde pueden apreciarse las evoluciones y las respectivas variaciones en las mediciones de los tiempos de ejecución en la medida que el tamaño de la matriz se incrementa. A medida que la dimensión de la matriz va aumentando, los tiempos entre CPU monoprocesador y GPU se asemejan. A modo resumido, se puede acotar que es ventajoso ejecutar este algoritmo en CPU monoprocesador para matrices de dimensiones inferiores a 300*300, en cambio, para aquellas de mayor dimensión será conveniente una arquitectura basada GPU la cual, como se puede visualizar, mejora notablemente los tiempos de respuesta.

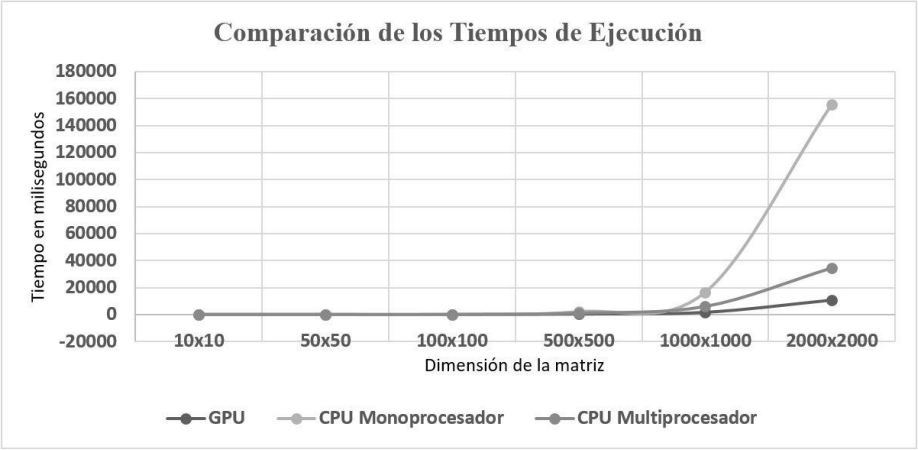

Fig. 4. Gráfico comparativo de los tiempos de ejecución del algoritmo de Barlow en las distintas arquitecturas aplicado a matrices de diferentes dimensiones.

Respecto a la precisión de los datos, al comenzar las mediciones a partir de las matrices de menor dimensión se observó, en algunos resultados, que cuando el algoritmo fue ejecutado en GPU existía una diferencia en el 13º decimal con respecto a las ejecuciones en CPU, tanto en monoprocesador como en multiprocesador, para las cuales los valores obtenidos fueron coincidentes. Sin embargo, a medida que la dimensión de la matriz crece, se detecta que esta diferencia comienza a incrementarse, por ejemplo, en matrices de dimensión 1000*1000 se encontraron diferencias en el 11º decimal. A partir de lo anterior, aunque las variaciones encontradas entre las

distintas implementaciones son ínfimas, teniendo en cuenta que las operaciones son las mismas, habría que investigar si los productos y las sumas en ambos tipos de procesadores son coincidentes, para de este modo evaluar la eficiencia del algoritmo en GPU.

5 Conclusiones

En el presente trabajo se presentó el desarrollo de un algoritmo alternativo que permite resolver el problema de la bidiagonalización de matrices densas y una comparación al implementarlo variando la arquitectura. El algoritmo fue probado adaptándolo a tres arquitecturas distintas: basada en GPU, CPU monoprocesador y multiprocesador. Se realizó un análisis de los tiempos resultantes para cada una de las implementaciones, observando la mejora en el rendimiento al paralelizar el algoritmo en una arquitectura basada en GPU para matrices de dimensiones mayores, cuando las matrices son pequeñas, en el orden de hasta 300*300, es preferible una arquitectura CPU monoprocesador. Es posible afirmar que las discrepancias en la precisión de los datos detectadas durante la ejecución sobre GPU son ínfimas, de todas maneras, sería necesario estudiar con mayor detalle los cálculos de cada procesador, incluso considerando matrices de mayor tamaño, para obtener conclusiones que ayuden a determinar si el comportamiento del algoritmo en dicha implementación se puede considerar exitosa.

En una siguiente etapa se pretende explorar si soluciones híbridas logran una aceleración en los cómputos, asignando en cada parte, CPU o GPU, aquellas tareas en las cuales mejor se desempeñan. Por otro lado, se espera poner a prueba la optimización conseguida en el SRI desarrollado por el equipo con el fin de comprobar el nivel de impacto alcanzado en la productividad del proceso.

Agradecimientos. Se agradece al Departamento de Ingeniería e Investigaciones Tecnológicas de la Universidad Nacional de La Matanza, el presente trabajo se financia en el marco del proyecto PROINCE C225.

Referencias

1. Deerwester, S., Dumais, S., Furnas, G., Landauer, T. & Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science. vol. 41(6), pp.391–407. 1990.
2. Mamani Roque, M.: Descomposición en Valores Singulares y Análisis Semántico Latente. Tesis de Maestría. Universidad Politécnica de Valencia, España, 2018.
3. Dong, T., Haidar, A., Tomov, S. & Dongarra, J.: Optimizing the SVD Bidiagonalization Process for a Batch of Small Matrices. Linear Algebra and Its Applications, ELSEVIER, vol. 108, pp. 1008-1018, 2017.
4. Ltaief, H., Luszczek, P., & Dongarra, J.: High performance bidiagonal reduction using tile algorithms on homogeneous multicore architectures. ACM Transactions on Mathematical Software, vol. 39(3), 2013.

5. Lahabar, S. & Narayanan, P.: Singular Value Decomposition on GPU using CUDA. IEEE International Symposium on Parallel & Distributed Processing, pp. 1-10, 2009.
6. Liu, F., Seinstra, F.: GPU-based parallel householder bidiagonalization. HPDC '10 Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, pp. 288-291, 2010.
7. Barlow, J., Bosner, N., Drmač, Z.: A new stable bidiagonal reduction algorithm. Linear Algebra and Its Applications, ELSEVIER, vol. 397, pp. 35-84, 2005.
8. Tolosa, G. & Bordignon, F.: Introducción a la Recuperación de Información: Conceptos, modelos y algoritmos básicos. Universidad Nacional de Luján, Argentina, 2008. Recuperado el 29/06/2020 de: http://eprints.rclis.org/12243/1/Introduccion-RI-v9f.pdf
9. Jaimes, L. & Riveros, F.: Modelos clásicos de recuperación de la información. Revista Integración. Escuela de Matemáticas. Universidad de Santander, vol. 23(1), pp. 17–26, 2005.
10. J. Demmel, M., Gu, S. Eisenstat, et al.: Computing the Singular Value Descomposition with Hight Relative Accuracy. Linear Algebra and its Application, vol. 299, pp. 21-80, 1999.
11. Berry, M., Dumais, S. & O'Brien, G.: Using Linear Algebra For Intelligent Information Retrieval. Society for Industrial and Applied Mathematics, Review, vol. 37(4), pp. 573-595. Philadelphia, USA, 1995.
12. Fortuna, L., Nunnari, G. & Gallo, A.: Model order reduction techniques with applications in electrical engineering. Springer-Verlag, 1992.
13. Ltaief, H., Kurzak, J. & Dongarra, J.: Parallel Two-Sided Matrix Reduction to Band Bidiagonal Form on Multicore Architectures. IEEE Transactions on Parallel and Distributed Systems, vol. 21(4), pp. 417 – 423, 2010.
14. Golub, G. & Reinsch, C.: Singular Value Decomposition and Least Squares Solutions, Handbook Series Linear Algebra, vol. 14, pp. 403-420, 1970.
15. Chan, T.: An Improved Algorithm for Computing the Singular Value Decomposition. ACM Transactions on Mathematical Software, vol. 8(1), pp. 72-83, 1982.
16. Sangwine, S. & Le Bihan, N.: Quaternion Singular Value Decomposition based on Bidiagonalization to a Real Matrix using Quaternion Householder Transformations. Applied Mathematics and Computation, ELSEVIER, 182(1): 727-738, 2006.
17. Da Silva Sanches de Campos, C.: Algoritmos de Altas Prestaciones para el Cálculo de la Descomposición en Valores Singulares y su Aplicación a la Reducción de Modelos de Sistemas Lineales de Control. Tesis Doctoral. Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, España, 2014.
18. Ralha, R.: One-sided reduction to bidiagonal form. Linear Algebra and Its Applications, ELSEVIER, 358(1-3): 219-238, 2003.
19. Guerrero López, D.: Algoritmos Paralelos para la Reducción de Sistemas Lineales de Control Estables. Tesis doctoral. Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, España, 2015.
20. Faverge M., Langou, J., Robert, Y. & Dongarra, J.: Bidiagonalization and R-Bidiagonalization: Parallel Tiled Algorithms, Critical Paths and Distributed-Memory Implementation. IEEE Transactions on Parallel and Distributed Processing Symposium, 668 - 677, 2017.
21. Hernández Cortés, J.: Implementación paralela y heterogénea de la transformación de Householder y sus aplicaciones. Tesis de Maestría. Departamento de Computación, Unidad Zacatenco, México, 2017.
22. Spositto, O., Ledesma, V. & Procopio, G.: Aplicación de la Descomposición de Valores Singulares a un Sistema de Recuperación de Información. Revista Digital del Departamento de Ingeniería (ReDDI). Universidad Nacional de La Matanza, vol. 4(2), 2019.

Implantación de un Algoritmo de Bidiagonalización en un Entorno Híbrido para su Aplicación en la Recuperación de Información

Osvaldo Spositto[1], Viviana Ledesma[1], Gastón Procopio[1], Victoria Saizar[1] y Alexis Vainberg[1]
[1]Universidad Nacional de La Matanza. Departamento de Ingeniería e Investigaciones Tecnológicas.
{spositto; vledesma; gprocopio; vsaizar; avainberg}@unlam.edu.ar

Resumen

La Indexación Semántica Latente es un método ampliamente utilizado en el contexto de la recuperación de información, para simular el análisis que realizaría una persona se utilizan algoritmos matemáticos especializados. Una de las técnicas empleadas a tal fin es la Descomposición en Valores Singulares. Esta técnica incluye una fase inicial en la que la matriz original es llevada a su forma bidiagonal, estudios indican que esta fase es la que insume el mayor tiempo del proceso, de ahí el interés en su optimización. En este trabajo se pretende evaluar si la adaptación de un algoritmo de bidiagonalización alternativo, al ser implementado en un entorno híbrido CPU-GPU, logra una aceleración en los cómputos. Se han comparado los resultados al implementar el mismo algoritmo en GPU. El estudio realizado permitió comprobar que la implementación en la arquitectura híbrida propuesta es una buena alternativa para matrices menores a una dimensión de 2000, identificando la necesidad de profundizar los estudios para matrices de mayores dimensiones.

1. Introducción

La Indexación Semántica Latente (ISL) es un método específico que proporcionó un salto importante para mejorar la precisión de la recuperación de información en documentos a través de la indexación de términos [1], esta surge en la década de los 80 buscando dar respuesta a tecnologías anteriores que no lograban entender la sinonimia y polisemia. La ISL utiliza una técnica llamada Descomposición en Valores Singulares (DVS), para posteriormente recuperar la información a partir de los valores y vectores singulares conseguidos mediante la aplicación de tal técnica [2].

En este artículo se presenta parte del trabajo realizado en el marco de un proyecto PROINCE, llevado adelante por investigadores del Departamento de Ingeniería e Investigaciones Tecnológicas de la Universidad Nacional de La Matanza. Su objetivo principal es obtener una mejora en la resolución de la DVS, para luego implementarla en un Sistema de Recuperación de Información (SRI), desarrollado por el mismo equipo de investigación. Específicamente el trabajo se relaciona con el desarrollo e implementación de algoritmos que permitan optimizar la primera fase del proceso de la DVS, la bidiagonalización [3]. Estudios realizados muestran que llevar la matriz inicial a su forma bidiagonal para obtener todos los vectores singulares o solo los valores singulares puede consumir entre un 70% o 90% del tiempo total del proceso, respectivamente [4].

Con lo anterior presente, se han analizado distintas variantes de algoritmos de bidiagonalización. En especial el foco se ha centrado en el desarrollo e implementación de un algoritmo alternativo propuesto por Barlow, Bosner y Drmač [5] (en adelante, Barlow), cuyas implementaciones están siendo estudiadas, como parte de la investigación, en la búsqueda de una reducción en sus tiempos de ejecución.

Dado que el método de bidiagonalización por sus características propias tiene un alto nivel de paralelismo en sus operaciones, se considera que la computación paralela y distribuida puede ser una alternativa eficaz para su solución. Así es que, en un estudio previo, el algoritmo de Barlow ha sido paralelizado, desarrollado e implementado sobre tres arquitecturas diferentes, monoprocesador, multiprocesador y GPU, con el fin de comparar el rendimiento en cada una de estas [6].

Con tal estudio como base, y continuando con el diseño, implementación y ejecución de soluciones paralelas, se ha puesto interés en un entorno híbrido entre procesador y unidad de procesamiento gráfico (CPU y GPU, respectivamente, por sus siglas en inglés). De este modo se pretende maximizar el aprovechamiento de las capacidades de cómputo que ofrece cada componente CPU o GPU, es decir, asignando a cada una de estas las tareas en las que mejor se desempeña.

Por lo tanto, este trabajo contiene un resumen de los resultados obtenidos a partir de la implementación del algoritmo en estudio en una arquitectura híbrida CPU-GPU y su comparación con la ejecución del mismo en una arquitectura basada sólo en GPU. Se desea destacar que se han encontrado trabajos en los que el algoritmo de Barlow es implementado y probado en MatLab, en CPU secuencial y paralelo, pero nunca ha sido probado en entornos con GPU. En este estudio la implementación se realizó utilizando el framework de CUDA para luego comparar los

resultados en base a los tiempos de respuesta cuando se aplica el algoritmo en matrices de distintos tamaños.

El resto del artículo se organiza de la siguiente manera: la Sección 2 explica resumidamente en qué consiste el problema que deben resolver los SRI y su relación con dos técnicas fuertemente asociadas a esta, la ISL y la DVS; la Sección 3 presenta el algoritmo de Barlow objeto de este estudio; la Sección 4 expone algunos avances en cuanto a la paralelización del método de bidiagonalización; la Sección 5 muestra los resultados experimentales obtenidos en el presente trabajo; y finalmente, la Sección 6 presenta las principales conclusiones e ideas para avanzar en esta investigación.

2. Técnicas para la Recuperación de Información

Los SRI involucran la representación, el almacenamiento, la organización y el acceso a los ítems de información [7]. En la Figura 1 se muestra de forma gráfica la problemática asociada a los SRI. Se dispone de una colección de documentos (corpus), por otra parte existen usuarios con necesidades de información que son planteadas al SRI en forma de consultas para que el mismo retorne como respuesta aquellos documentos considerados relevantes, por satisfacer la necesidad expresada. Estos se devuelven en forma de una lista ordenada (rankeada).

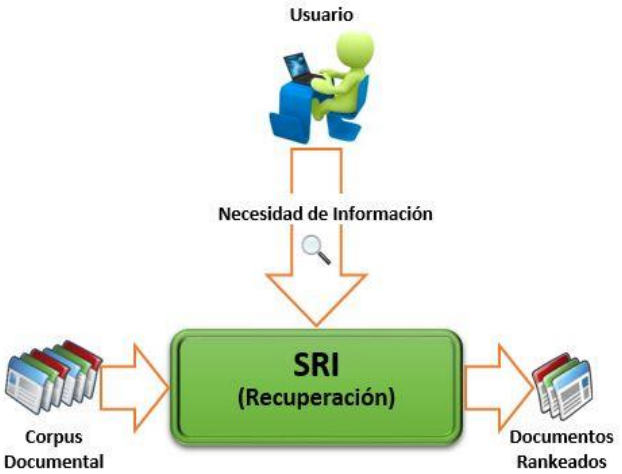

Figura 1. Problemática de los SRI

Para realizar operaciones sobre un corpus, se necesita primero una representación lógica de todos sus documentos como también de las consultas con el fin de comparar la similitud entre tales representaciones. Con tal propósito, conviven actualmente una variedad de modelos que se apoyan en distintos paradigmas [8]. Se destacan entre estos, los modelos denominados clásicos: el modelo booleano, el modelo vectorial y, el modelo probabilístico. En los distintos modelos se seleccionan las palabras útiles, que por lo general son todos los términos del documento a excepción de las palabras sincategoremáticas, es decir, semánticamente sin significado, este proceso se enriquece utilizando técnicas de lematización y etiquetado [9]. En particular, el modelo vectorial representa las consultas y documentos mediante vectores [10]. Así, cada documento se ubica dentro de un espacio vectorial determinado según los términos que contiene. Cada término de un documento está representado por los pesos de los términos contenidos en el mismo.

El presente trabajo está asociado a una extensión del modelo de recuperación vectorial, la ISL. Este método permite la búsqueda de información en documentos mediante la indexación de sus términos [1]. Comprende establecer un espacio semántico donde los términos y los documentos fuertemente relacionados son colocados unos cerca de otros, mostrando de este modo los patrones de asociación entre los datos más importantes e ignorando los que tienen menor influencia al momento de la recuperación.

La aplicación de la ISL, como se indicó anteriormente, implica la utilización de algoritmos matemáticos especializados. Permite buscar por conceptos o definiciones en contraste a lo que sería una búsqueda literal. Pretende solucionar problemas de sinonimia y polisemia, o equivocidad del habla corriente. Para tal fin, un primer recurso es trabajar con lexemas y no con palabras, ya que palabras derivadas de una misma raíz comparten buena parte de la carga semántica.

Generalmente, la indexación de los términos de los documentos, da por resultado matrices de documentos que se vuelves de grandes dimensiones. Por tal razón, en la búsqueda de acelerar el proceso de recuperación de información, suelen aplicarse técnicas de reducción de la dimensionalidad con el fin de transformar dicha matriz en una de menores dimensiones, pero capaz de reflejar las características de la matriz original al momento de procesar las búsquedas. Con tal propósito, se aplica la DVS, una técnica de factorización de matrices mediante la cual se descompone una matriz en varias matrices que presentan las propiedades más significativas de la matriz original [1, 11, 12]. Así, como se muestra en la Figura 2, una matriz A de tamaño t × d descompuesta con DVS produce tres matrices, de la forma: $A = T_0 \times S_0 \times D_0$.

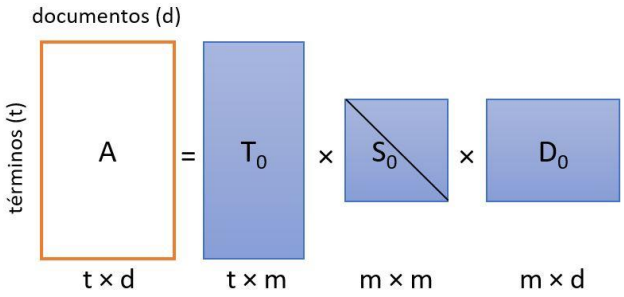

Figura 2. Reducción de las dimensiones en la DVS. Fuente: [1]

Las columnas de T_0 son ortonormales entre sí y en D_0 lo son las filas, son las matrices izquierda y derecha respectivamente, de vectores singulares y, S_0 es una matriz diagonal compuesta de los valores singulares de A. El triple producto indicado da una matriz de $t \times d$ de rango m. De todas las matrices de $t \times d$ de rango m que aproximen a A, la de menor error, es decir distancia, es una que comparte los mayores m autovalores de A, obtenidos en una

descomposición DVS y anula los restantes, comparte sus autovectores. Habiendo autovalores nulos sus correspondientes componentes en los autovalores no tienen influencia y por lo tanto son recortados a tamaño *m,* siendo *m* un valor elegido como compromiso entre ahorro de memoria y precisión que da un valor máximo con el cual la reconstrucción es perfecta.

Estos modelos de orden reducido, según [13], presentan las siguientes ventajas:

- simplifican la comprensión del sistema,
- reducen el coste computacional en los problemas de simulación, lo cual a su vez implica menor esfuerzo computacional en el diseño de controladores numéricamente más eficientes, y
- se obtienen leyes de control más simples.

Lo anterior justifica la importancia y necesidad de conseguir modelos matemáticos simplificados que aproximen al máximo el comportamiento del sistema original. Existen dos tipos principales de algoritmos dedicados al cálculo computacional de la DVS de una matriz real, el método unilateral de Jacobi y los métodos basados en la bidiagonalización [11]. El número de operaciones para los distintos algoritmos se encuentra en el orden de $O(n^3)$, las diversas propuestas y mejoras que han surgido buscan disminuir operaciones costosas en tiempo. El trabajo que viene llevando adelante este equipo de investigación se enmarca en los algoritmos basados en bidiagonalización, en los que se aplican transformaciones ortogonales con el fin de obtener una forma bidiagonal para luego conseguir la DVS de la matriz bidiagonal.

3. Un Algoritmo Alternativo para la Bidiagonalización

La reducción bidiagonal de una matriz densa general, como se indicó previamente, se aplica muy frecuentemente como fase preliminar para el cálculo de la DVS [14]. En la literatura se han encontrado distintos métodos para la bidiagonalización de una matriz, las visiones más tradicionales se basan en las transformaciones de Householder por la izquierda y por la derecha de la matriz [11, 15, 16]. Diversos estudios demuestran ciertas desventajas en dichos métodos, los tiempos cúbicos son de preocupar cuando las matrices son grandes y por otra parte, repercuten negativamente en los costos de comunicación de una implementación paralela del algoritmo en sistemas de memoria distribuida [17, 18].

Así es que han surgido diversos trabajos, entre estos se encuentran la propuesta de Ralha [19], luego mejorada por Barlow [5], que se dirigen a conseguir un método más sencillo de paralelizar que los métodos tradicionales. La particularidad de su algoritmo es que la bidiagonalización es unilateral, donde sólo se aplican las transformaciones de Householder por el lado derecho de la matriz. En esta misma línea, Da Silva Sanches de Campos [18] propuso una optimización al método de Barlow con el objetivo de reducir el número de comunicaciones que se necesitan para una implementación paralela destinada a sistemas de memoria distribuida.

El método de Barlow se puede expresar, en forma de algoritmo, como se muestra en la Figura 3. Este consiste en un ciclo en el que se trabaja la matriz principal por columnas, va desde la columna *0* hasta la antepenúltima columna *(n-2)*. Además de la matriz inicial, hay 3 variables principales que se utilizan a lo largo de todo el algoritmo:

- α: es un vector de *n* elementos que contiene los valores de la diagonal principal.
- β: es un vector de *n-1* elementos que contiene los valores de la diagonal superior.
- *q*: es una matriz con idénticas dimensiones que la matriz principal, es una matriz de trabajo interno.

En cada iteración se va completando: una posición en el vector α de elementos de la diagonal principal; una posición en el vector β de elementos de la diagonal superior, esto se representa en las líneas 5 y 6 del algoritmo en las que se aplican las reflexiones de Householder; una columna de la matriz *q*; y además, se modifica la matriz inicial que luego se lee en las iteraciones subsiguientes. Al terminar el ciclo se completan las posiciones restantes de α, β y las columnas que quedan de la matriz *q*.

Algoritmo **1 :** BarlowBidiagonalización (A, α, β, Q)

1 **for** $r = 1, 2, ..., n-2$ **do**
2 $\quad \alpha_r = ||A(:,r)||_2$
3 $\quad q_r = \dfrac{A(:,r)}{\alpha_r}$
4 $\quad x_r = \mathrm{A}\ (:, r+1:n)^t\ q_r$
5 $\quad H_r$ tal que $H_r^t\ x_r = \beta_r\ e_1$
6 $\quad \mathrm{A}(:, \mathrm{r}+1:\mathrm{n}) = \mathrm{A}\ (:, \mathrm{r}+1:\mathrm{n})\ H_r$
7 $\quad \mathrm{A}(:, \mathrm{r}+1) = \mathrm{A}\ (:, \mathrm{r}+1) - \beta_r\ q_r$
8 **end**
9 $\alpha_{n-1} = ||A(:,n-1)||_2$
10 $q_{n-1} = \dfrac{A(:,n-1)}{\alpha_{n-1}}$
11 $\beta_{n-1} = q_{n-1}^t\ \mathrm{A}\ (:, \mathrm{n})$
12 $\mathrm{A}(:, \mathrm{n}) = \mathrm{A}\ (:, \mathrm{n}) - \beta_{n-1} q_{n-1}$
13 $\alpha_n = ||A(:,n)||_2$
14 $q_n = \dfrac{A(:,n)}{\alpha_n}$

Figura 3. Algoritmo de bidiagonalización unilateral de Barlow. Fuente: [18]

El hecho de aplicar Householder unilateralmente permite definir todas las operaciones en términos de las columnas de la matriz a transformar, posibilita el desarrollo de implementaciones paralelas de un modo más simplificado en comparación con los métodos tradicionales, por otra parte, se logra reducir las comunicaciones que se necesitan.

4. Paralelización de la Bidiagonalización

Las operaciones utilizadas para llevar adelante el método de bidiagonalización hacen que su proceso sea altamente paralelizable [20]. Sobra aclarar, que la correcta ejecución de algoritmos paralelos está fuertemente asociada a que los tamaños de las matrices se adapten a las capacidades del hardware donde estos se ejecutan, por lo que, en matrices de alta dimensionalidad, aparecen problemas como el espacio en la memoria, el incremento en los tiempos de ejecución, así también, los algoritmos para tamaños grandes deben ser modificados ya que no operan sobre columnas completas de una vez, lo que exige análisis de correctitud más tediosos.

Durante la revisión en la literatura se han encontrado varios trabajos orientados a estudios comparativos sobre el rendimiento al bidiagonalizar matrices de distintos tamaños cuando se utilizan distintas implementaciones variando la arquitectura. Algunos han realizado estudios para contrastar implementaciones secuenciales y paralelas sobre una arquitectura homogénea basada en CPU [18], se han experimentado algoritmos en mosaico con distinta cantidad de nodos multinúcleo de un sistema de memoria compartida distribuida en paralelo [4], [21]. Otros han buscado aprovechar la capacidad que ofrecen las GPU y experimentaron su uso aplicando algoritmos en arquitecturas homogéneas [3, 22] como también heterogéneas en las que se combinan el uso de CPU con GPU [23].

Los modelos de programación orientados a la computación en paralelo cambian sensiblemente y requieren de nuevos recursos. Entre las diversas soluciones de computación en paralelo disponibles, las GPU resultan de gran utilidad cuando se deben procesar grandes volúmenes de datos en paralelo, con los que se consiguen importantes mejoras en cuanto a rendimiento a bajo costo. Dada su elevada capacidad de cálculo, cada vez son más utilizadas, no solamente para la generación de gráficos sino también para el cómputo de algoritmos de propósito general como en el álgebra lineal, algoritmos de ordenación búsqueda y procesamiento de consultas, entre otros, logrando una gran mejora en el speedup de la ejecución de las aplicaciones asociadas [24]. La utilización de la potencia de cómputos que brindan las GPU en aplicaciones de propósito general para obtener un alto rendimiento ha dado lugar al concepto de GPGPU (General-Purpose Computing on Graphics Processing Unit). Las GPU tienen una arquitectura SIMD (por sus siglas en inglés Single Instruction, Multiple Data), en estas se manipulan datos de tipos vectoriales a los cuales se aplican operaciones vectoriales [25].

Siendo que el algoritmo propuesto por Barlow en [5], como se indicó previamente, está pensado para soportar el paralelismo, este equipo de investigación ha decidido poner especial interés en su adaptación para implementaciones en arquitecturas basadas en GPU, y de esta manera, al obtener mejores tiempos de ejecución, se estaría consiguiendo consecuentemente un nuevo avance en la DVS.

En consonancia con lo anterior, se ha realizado un estudio en el cual el algoritmo ha sido paralelizado y desarrollado para ser implementado en GPU y los resultados obtenidos en cuanto a rendimiento se compararon con los obtenidos en arquitecturas monoprocesador, multiprocesador, este trabajo se describe parcialmente en [6].

En la Figura 4 pueden apreciarse las evoluciones y las respectivas variaciones en las mediciones de los tiempos de ejecución, para cada una de las arquitecturas mencionadas, en la medida que el tamaño de la matriz se incrementa. Cuando la dimensión de la matriz va aumentando, los tiempos entre CPU monoprocesador y GPU se asemejan.

De los tiempos obtenidos, como resultado de las pruebas, se observa que cuando las matrices son de menor dimensión es conveniente ejecutar este tipo de algoritmos en CPU monoprocesador. En cambio, cuando la matriz comienza a superar las dimensiones, aproximadamente a partir de un orden de 200 o 300, la GPU mejora notoriamente el tiempo de ejecución con respecto a CPU monoprocesador y CPU multiprocesador. Esto se puso en evidencia, por ejemplo, observando los tiempos insumidos para la matriz de orden 2000, donde la GPU logró reducir los tiempos de ejecución en un 93% y 69%, con relación a CPU monoprocesador y multiprocesador respectivamente. En cuanto a la CPU multiprocesador, se puede observar que es constante el tiempo de resolución y este se incrementa lentamente hasta las dimensiones aproximadas a 200.

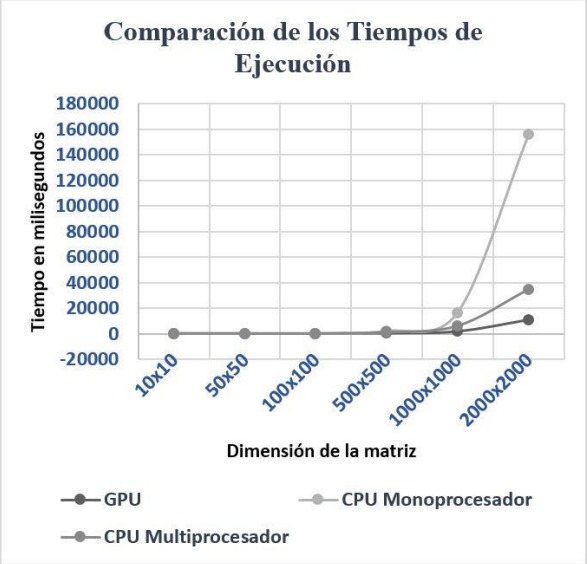

Figura 4. Gráfico comparativo de los tiempos de ejecución del algoritmo de Barlow en las distintas arquitecturas aplicado a matrices de diferentes dimensiones

A modo resumido, con este estudio inicial se llegó a la conclusión de que para aquellas de mayor dimensión será conveniente una arquitectura basada GPU la cual, como se puede visualizar, mejora notablemente los tiempos de respuesta, en contrapartida, es ventajoso ejecutar este algoritmo en CPU monoprocesador para matrices de dimensiones inferiores a 200. La razón de esto es que la implementación basada en GPU requiere de una

transferencia de la matriz de la RAM a la memoria de la GPU y, luego de su resolución el retorno, lo que puede comprender un gran porcentaje del tiempo de ejecución.

Además, la CPU está compuesta por núcleos complejos, lo cual optimiza las ejecuciones posteriores, mientras que la GPU son núcleos simples. La ventaja de la GPU cuando las dimensiones son mayores proviene de la cantidad de núcleos y no por ser superior en el modo de hacer la operación, de ahí también su pérdida en la precisión.

5. Implementación del Algoritmo en un Entorno Híbrido

Las implementaciones del algoritmo de Barlow han sido desarrolladas utilizando el lenguaje C#, en conjunto con el framework CUDA, versión 6.5.

Las características del equipo utilizado para las pruebas de este experimento son:

- CPU: AMD Ryzen 5 2600 6 núcleos 12 threads a 3.6 GHz
- Memoria: 2 x 8GB DDR4 Crucial Ballistix 2400 Mhz
- GPU: NVIDIA GEFORCE GTX 1050 2GB

Para las pruebas se utilizaron matrices cuadradas de orden *n,* donde *n* va desde 200 a 2000, en escalas de 100 en 100. Tomando como fundamento las conclusiones aportadas por Da Silva Sanches de Campos en [18], estas se componen de valores aleatorios, considerando que estos no tienen incidencia en los resultados esperados.

Como paso inicial, el algoritmo de bidiagonalización de Barlow ha sido desarrollado para funcionar de modo independiente tanto en CPU como en GPU.

Debe tenerse en cuenta que en GPU las matrices son tratadas de manera vectorial dado que esta no interpreta arrays bidimensionales o superiores. Esto conllevó a que el código del algoritmo a utilizar en CPU debiera ser modificado para dar un tratamiento a las matrices de igual manera.

Con base en el estudio anterior, tal como se mencionó previamente, cuando las dimensiones de las matrices son inferiores 200, se obtienen tiempos de ejecución menores si el algoritmo se ejecuta en CPU, mientras que a medida que crece el tamaño de la matriz resulta más ventajosa la implementación en GPU.

La idea principal de esta arquitectura híbrida es aprovechar la capacidad de cálculo de cada una de estas implementaciones, en lo que respecta a tiempos de ejecución, identificando las columnas de la matriz que conviene procesar en CPU y cuáles en GPU.

De esta manera se espera analizar si es posible obtener mejoras en los tiempos insumidos con respecto a la ejecución completa en GPU.

5.1. Definición de Cotas

La propuesta de este enfoque híbrido, como se indicó antes, implica inicialmente identificar cuál sería el punto de quiebre o cota, para definir qué columnas de la matriz se deberían procesar en la GPU, dejando las restantes para ser ejecutadas en la CPU.

Siendo que los resultados arrojaron que la CPU procesa más rápido hasta $n <= 200$, lo que representa unos 40000 elementos, se optó inicialmente por tomar esta medida como referencia, utilizándola para calcular las columnas necesarias para contener una cantidad de elementos igual o inferior a 40000, esto es:

$$\text{Cota 1} = 40000 / n$$

Debe tenerse en cuenta que para la implementación que se propone y evalúa en este trabajo, el pasaje de la matriz a GPU, se realiza siempre al inicio, lo cual tiene su costo. Para matrices de $n > 200$ conviene afrontar el mismo, sin embargo, la idea fue sacar ventaja del desempeño de la CPU enviando parte del trabajo a esta, precisamente para aprovecharla en lo que mejor se desempeña. Esto es, utilizando la Cota 1 y enviar esa cantidad de columnas para ser procesadas en la CPU.

Al aplicar la Cota 1 se observó que si la cantidad de columnas enviadas a la CPU es mayor a las procesadas en GPU se estaría produciendo un desaprovechamiento con el hecho de haber movido la matriz a la GPU. Con lo cual se decidió probar con una cota que permita distribuir mejor el trabajo, y analizar qué resultados se obtendrían limitándose a la mitad de los elementos, es decir 20000, así se definió:

$$\text{Cota 2} = 20000 / n$$

Los tiempos de ejecución en milisegundos obtenidos en la prueba al utilizar cada una de estas cotas se detallan en la Tabla 1.

Tabla 1. Tiempos de ejecución al variar el tamaño de la cota

Dimensión de la matriz	Cota 1	Cota 2
200	107	**66**
300	124	**101**
400	175	**159**
500	261	**251**
600	387	**382**
700	656	**550**
800	**950**	963
900	**1261**	1281
1000	**1615**	1629
1100	**2071**	2084
1200	**2586**	2615
1300	**3193**	3236
1400	**3879**	3923
1500	**4672**	4738
1600	**5555**	5607
1700	**6634**	6708
1800	**7755**	7838
1900	**9027**	9113
2000	**10396**	10491

Como se ha resaltado en la Tabla 1, se puede observar que la ejecución llevó menos tiempo cuando se utilizó la Cota 2, pero sólo en las primeras seis matrices, es decir, hasta la matriz de dimensión $n = 700$. A partir de $n = 800$ se consiguieron mejores resultados cuando se utilizó la Cota 1.

Siendo n la dimensión de la matriz, mientras que a y b son constantes. La fórmula está pensada de tal manera que tenga una asíntota horizontal en el valor al que debe tender el resultado a medida que n crece. Esta asíntota debería estar en 20, la cota óptima para $n = 2000$. Por lo tanto, el primer término a debe ser 20, ya que el segundo término tiende a 0 cuando n crece.

El valor de b se definió de forma que cuando n sea 200, el menor tamaño de matriz a ejecutarse en GPU, la cota valga 100 (la cota óptima para esa dimensión), para ello, el segundo término debería valer 80:

$$b / 200 = 80$$

Con lo cual, despejando b se obtiene como resultado que su valor es 16.000. Es así que la fórmula finalmente puede expresarse como:

$$\text{Cota 3} = 20 + 16000 / n.$$

En la Figura 5 se puede visualizar una representación lineal del comportamiento de la curva asociada a cada cota, en lo referido a la cantidad de columnas de la matriz que deberán ser enviadas para su procesamiento en CPU. Tal como se puede visualizar, con la fórmula propuesta para la Cota 3, se obtuvo un comportamiento similar a la Cota 2 para matrices de dimensiones entre 200 y 700, mientras que se acerca a la Cota 1 para aquellas cuya dimensión se encuentra entre 700 y 2000.

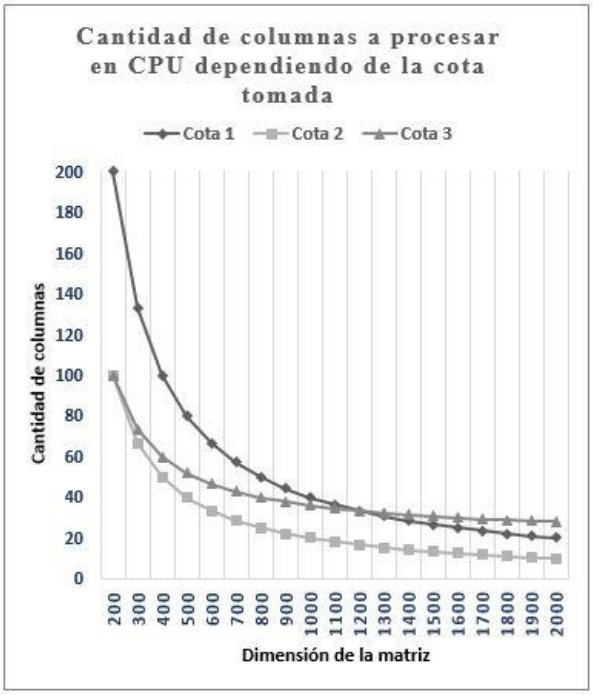

Figura 5. Comparación de la cantidad de columnas a ser procesadas por la CPU dependiendo de la cota utilizada

En la Figura 6 se muestran los resultados del experimento en cuanto a tiempos en milisegundos insumidos para bidiagonalizar cada matriz, incluyendo en este caso la comparación cuando se utilizan cada una de las tres cotas analizadas.

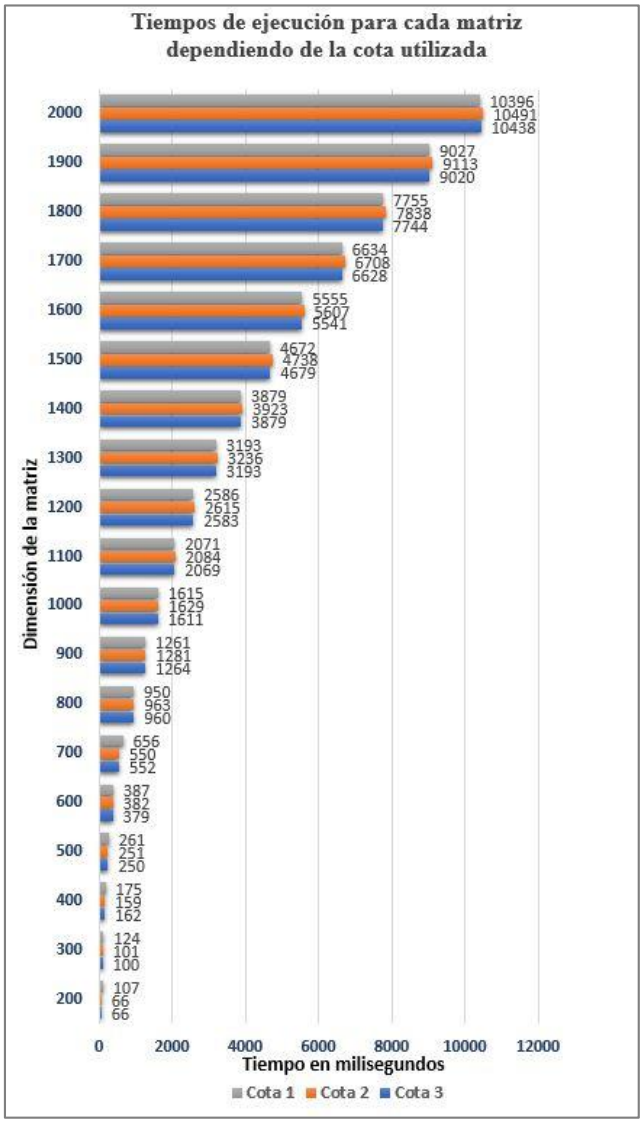

Figura 6. Gráfico comparativo de los tiempos de obtenidos al utilizar cada cota

Observando detalladamente los valores de los tiempos de ejecución mostrados en la Figura 6, al utilizar la Cota 3, tal como se esperaba, se ha logrado aproximar los tiempos a los mejores resultados alcanzados tanto para los casos en que el menor tiempo se obtuvo con la Cota 1, como también si se utilizó la Cota 2.

Incluso, en un 68% de las matrices utilizadas en el experimento, se ha conseguido una leve mejora con respecto a los menores tiempos conseguidos entre las cotas 1 y 2, disminuyendo el tiempo total de bidiagonalización hasta en 14 milisegundos según el valor de n.

Sin embargo, también se observa una notoria diferencia para la matriz de $n = 2000$ donde la utilización de la Cota 3 insumió un tiempo de ejecución de 69 milisegundos de por encima del mejor tiempo alcanzado antes. En la Tabla 2 es

posible visualizar con más detalle los resultados de ejecución utilizando la Cota 3 y su comparación con los mejores tiempos obtenidos en la prueba anterior usando la Cota 1 y la Cota 2.

Tabla 2. Comparación de tiempos de ejecución utilizando la Cota 3 con respecto al menor tiempo obtenido

Dimensión de la matriz	Menor tiempo (Cota 1 o 2)	Cota 3
200	66	**66**
300	101	**100**
400	159	162
500	251	**250**
600	382	**379**
700	550	552
800	950	960
900	1261	1264
1000	1615	**1611**
1100	2071	**2069**
1200	2586	**2583**
1300	3193	**3193**
1400	3879	**3879**
1500	4672	4679
1600	5555	**5541**
1700	6634	**6628**
1800	7755	**7744**
1900	9027	**9020**
2000	10396	10438

Por otra parte, en la Figura 7 se refleja el porcentaje de columnas de la matriz que fueron asignadas a la CPU para ser procesadas, cuando se utilizó la Cota 3. A medida que el tamaño de la matriz aumenta la mayor parte del proceso se ejecuta en la GPU.

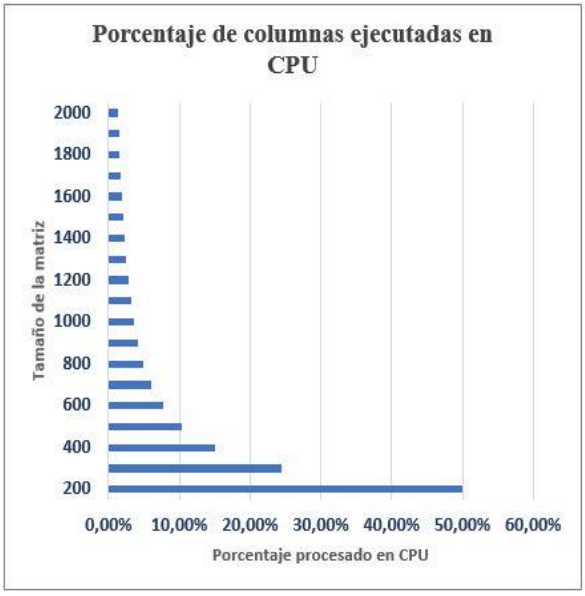

Figura 7. Porcentaje de columnas procesadas en la CPU con una cota de n=16000

5.2. Comparación de la arquitectura híbrida con la implementación en GPU

Una vez conseguidos los tiempos de ejecución con la implementación híbrida, se procedió a comparar los resultados con respecto a los tiempos insumidos al bidiagonalizar las mismas matrices pero haciendo el proceso completo en GPU. En la Tabla 3 se presentan los resultados obtenidos en milisegundos, lo mismo se representa en un gráfico lineal en la Figura 8.

Tabla 3. Tiempos de ejecución insumidos en la arquitectura híbrida CPU-GPU y en GPU

Dimensión de la matriz	Arquitectura Híbrida	GPU
200	**66**	68
300	**100**	112
400	**162**	169
500	**250**	253
600	**379**	389
700	**552**	679
800	**960**	971
900	**1264**	1286
1000	**1611**	1635
1100	**2069**	2093
1200	**2583**	2611
1300	**3193**	3219
1400	**3879**	3903
1500	**4679**	4713
1600	**5541**	5578
1700	**6628**	6659
1800	**7744**	7768
1900	**9020**	9058
2000	10438	10430

Como se puede apreciar en los tiempos obtenidos, al comparar ambas implementaciones, se logró una reducción de tiempo que va de 2 a 127 milisegundos, dependiendo de la dimensión de la matriz, a favor de la arquitectura híbrida con respecto a GPU.

Sin embargo, en la Tabla 3 también puede observarse que cuando n es igual a 2000 el tiempo de ejecución fue mínimamente más favorable para la implementación en GPU, la cual muestra una ventaja de 8 milisegundos. Esto hace suponer que para matrices con $n > 2000$ podría ser conveniente aplicar una arquitectura basada únicamente en GPU, de todos modos para generalizar esta afirmación sería necesario repetir esta prueba con matrices de mayores dimensiones.

Por lo tanto, los resultados conseguidos con la implementación híbrida demuestran la necesidad de avanzar en este experimento, un camino posible sería replicar las pruebas con matrices de mayores dimensiones, buscando la mejor cota para cada tamaño, para luego identificar la función que se aproxime a tales resultados, suponiendo una función unimodal.

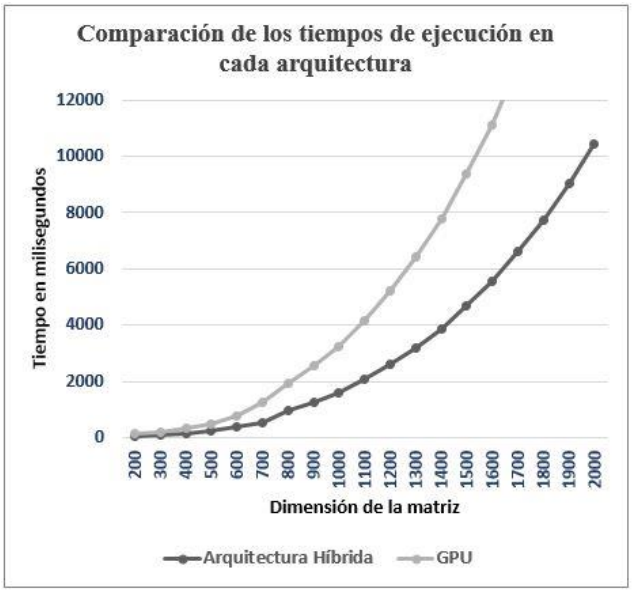

Figura 8. Gráfico lineal comparativo entre la implementación en arquitectura híbrida y GPU

6. Conclusión

En este trabajo se ha presentado el desarrollo de un algoritmo alternativo que permite resolver el problema de la bidiagonalización de matrices densas y una comparación al implementarlo en dos arquitecturas distintas.

Primeramente, el algoritmo fue adaptado para ser ejecutado en una arquitectura híbrida CPU-GPU, esto implico identificar la cota más adecuada que permitiera asignar a cada parte aquellas columnas para las cuales presenta un mejor desempeño con relación a los tiempos de ejecución. Como parte de este mismo estudio, también se puso interés en bidiagonalizar las mismas matrices, pero en este caso implementando el algoritmo en una arquitectura basada sólo en GPU.

Se evaluaron los rendimientos alcanzados por cada una de las implementaciones realizadas al procesar matrices de distintas dimensiones, observando en general una mínima mejora en cuanto a los tiempos conseguidos cuando el algoritmo fue paralelizado en una arquitectura híbrida. También pudo notarse, que para una matriz con una dimensión de 2000, el comportamiento en GPU fue más eficiente. Sin embargo, en cuanto a esto, sería necesario replicar estas pruebas con matrices de mayores dimensiones para obtener resultados que puedan ser generalizados, incluso se debería analizar la posibilidad de una optimización en lo referido a la cota utilizada.

Como siguiente paso, se espera poner a prueba la optimización conseguida en el SRI desarrollado por este mismo equipo con la finalidad de comprobar el nivel de impacto alcanzado en la productividad del proceso.

7. Referencias

[1] Deerwester, S., Dumais, S., Furnas, G., Landauer, T. y Harshman, R., "Indexing by latent semantic analysis", Journal of the American Society for Information Science, Septiembre 1990, vol. 41(6), pp.391–407.

[2] Mamani Roque, M., "Descomposición en Valores Singulares y Análisis Semántico Latente", Universidad Politécnica de Valencia, España, Tesis de Maestría, 2018.

[3] Dong, T., Haidar, A., Tomov, S. y Dongarra, J., "Optimizing the SVD Bidiagonalization Process for a Batch of Small Matrices", en International Conference on Computational Science, ICCS 2017, 12-14 Junio 2017, Zúrich, Suiza, vol. 108, pp. 1008-1018.

[4] Ltaief, H., Luszczek, P., y Dongarra, J., "High performance bidiagonal reduction using tile algorithms on homogeneous multicore architectures", en ACM Transactions on Mathematical Software, 2013, vol. 39(3).

[5] Barlow, J., Bosner, N. y Drmač, Z., "A new stable bidiagonal reduction algorithm", en Linear Algebra and Its Applications, ELSEVIER, 2005, vol. 397, pp. 35-84.

[6] Spositto, O., Ledesma, V., Procopio, G., Ryckeboer, H., Saizar, V., Vainberg, A., "Comparación de un Algoritmo de Bidiagonalización para su Utilización en la Recuperación de Información", "enviado para evaluación" a CACIC 2020.

[7] Baeza-Yates, R. y Ribeiro-Neto, B, "Modern Information Retrieval", USA, Addison Wesley, 1999.

[8] Tolosa, G. y Bordignon, F., "Introducción a la Recuperación de Información: Conceptos, modelos y algoritmos básicos", Universidad Nacional de Luján, Argentina, 2008. Disponible en: http://eprints.rclis.org/12243/1/Introduccion-RI-v9f.pdf. Consultado el 12/08/2020.

[9] Jaimes, L. y Riveros, F., "Modelos clásicos de recuperación de la información", en Revista Integración, Escuela de Matemáticas, Universidad de Santander, 2005, vol. 23(1), pp. 17–26.

[10] Salton, G., Wong, A. y Yang, C., "A vector space model for automatic indexing", en Communications of the ACM 18, 1975, Nr. 11, p. 613-620.

[11] Demmel,J., Gu, M., Eisenstat, S. et al., "Computing the Singular Value Descomposition with Hight Relative Accuracy", en Linear Algebra and Its Applications, ELSEVIER, 1999, vol. 299, pp. 21-80.

[12] Berry, M., Dumais, S. y O'Brien, G., "Using Linear Algebra For Intelligent Information Retrieval", Society for Industrial and Applied Mathematics, Philadelphia, USA, 1995, Review, vol. 37(4), pp. 573-595.

[13] Fortuna, L., Nunnari, G. y Gallo, A., "Model order reduction techniques with applications in electrical engineering", Springer-Verlag, 1992.

[14] Ltaief, H., Kurzak, J. y Dongarra, J., "Parallel Two-Sided Matrix Reduction to Band Bidiagonal Form on Multicore Architectures", en IEEE Transactions on Parallel and Distributed Systems, 2010, vol. 21(4), pp. 417 – 423.

[15] Golub, G. y Reinsch, C. "Singular Value Decomposition and Least Squares Solutions", Handbook Series Linear Algebra, 1970, vol. 14, pp. 403-420.

[16] Chan, T., “An Improved Algorithm for Computing the Singular Value Decomposition”, en ACM Transactions on Mathematical Software, 1982, vol. 8(1), pp. 72-83.

[17] Sangwine, S. y Le Bihan, N., “Quaternion Singular Value Decomposition based on Bidiagonalization to a Real Matrix using Quaternion Householder Transformations”, en Applied Mathematics and Computation, ELSEVIER, 2006, 182(1), pp. 727-738.

[18] Da Silva Sanches de Campos, C., “Algoritmos de Altas Prestaciones para el Cálculo de la Descomposición en Valores Singulares y su Aplicación a la Reducción de Modelos de Sistemas Lineales de Control”, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, España, Tesis Doctoral, 2014.

[19] Ralha, R., “One-sided reduction to bidiagonal form”, en Linear Algebra and Its Applications, ELSEVIER, 2003, 358(1-3), pp. 219-238.

[20] Guerrero López, D., “Algoritmos Paralelos para la Reducción de Sistemas Lineales de Control Estables, Departamento de Sistemas Informáticos y Computación”, Universidad Politécnica de Valencia, España, Tesis Doctoral, 2015.

[21] Faverge M., Langou, J., Robert, Y. y Dongarra, J., “Bidiagonalization and R-Bidiagonalization: Parallel Tiled Algorithms, Critical Paths and Distributed-Memory Implementation”, en IEEE Transactions on Parallel and Distributed Processing Symposium, Orlando, Florida, 2017, pp. 668 – 677.

[22] Lahabar, S. y Narayanan, P., “Singular Value Decomposition on GPU using CUDA”, en IEEE International Symposium on Parallel & Distributed Processing, Roma, 2009, pp. 1-10.

[23] Hernández Cortés, J., “Implementación paralela y heterogénea de la transformación de Householder y sus aplicaciones”, Departamento de Computación, Unidad Zacatenco, México, Tesis de Maestría, 2017.

[24] Guim, F. y Rodero, I., “Arquitecturas basadas el Computación Gráfica (GPU)”, Universitat Overta de Catalunya. PID_00184818. Disponible en: http://repositorio.itsjapon.edu.ec:90/jspui/handle/123456789/400. Consultado el 11/08/2020.

[25] Piccoli, M., “Computación de alto desempeño en GPU”, Journal of Computer Science & Technology, La Plata, Argentina, 2012, Disponible en: htt/://sedici.unlp.edu.ar/handle/10915/18404. Consultado el 10/08/2020.

ANEXO I

COPIA DE CERTIFICADOS

Se certifica que **HUGO EMILIO RYCKEBOER (UNLAM)** ha participado en calidad de autor del artículo **HACIA LA OPTIMIZACIÓN DE UN SISTEMA DE RECUPERACIÓN DE INFORMACIÓN (12744 - PDP)** aceptado en el XXII WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN – WICC 2020, organizado por la Universidad Nacional de la Patagonia Austral - Junio 2020.

Lic. Patricia Pesado
Coordinadora
RedUNCi

Ing. Hugo Santos ROJAS
Rector
UNPA

Se certifica que **OSVALDO MARIO SPOSITTO (UNLAM)** ha participado en calidad de autor del artículo **HACIA LA OPTIMIZACIÓN DE UN SISTEMA DE RECUPERACIÓN DE INFORMACIÓN (12744 - PDP)** aceptado en el XXII WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN – WICC 2020, organizado por la Universidad Nacional de la Patagonia Austral - Junio 2020.

Lic. Patricia Pesado
Coordinadora
RedUNCi

Ing. Hugo Santos ROJAS
Rector
UNPA

Se certifica que **VIVIANA LEDESMA (UNLAM)** ha participado en calidad de autor del artículo **HACIA LA OPTIMIZACIÓN DE UN SISTEMA DE RECUPERACIÓN DE INFORMACIÓN (12744 - PDP)** aceptado en el XXII WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN – WICC 2020, organizado por la Universidad Nacional de la Patagonia Austral - Junio 2020.

Lic. Patricia Pesado
Coordinadora
RedUNCi

Ing. Hugo Santos ROJAS
Rector
UNPA

Se certifica que **GASTÓN PROCOPIO (UNLAM)** ha participado en calidad de autor del artículo **HACIA LA OPTIMIZACIÓN DE UN SISTEMA DE RECUPERACIÓN DE INFORMACIÓN (12744 - PDP)** aceptado en el XXII WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN – WICC 2020, organizado por la Universidad Nacional de la Patagonia Austral - Junio 2020.

Lic. Patricia Pesado
Coordinadora
RedUNCi

Ing. Hugo Santos ROJAS
Rector
UNPA

Congreso Argentino de Ciencias de la Computación

San Justo, Octubre de 2020

Se certifica que

Osvaldo Mario Spositto, Viviana Ledesma, Gastón Procopio, Hugo Emilio Ryckeboer, Victoria Saizar y Alexis Vainberg

han participado como Autores del artículo 13380 *"Comparación de un Algoritmo de Bidiagonalización para su Utilización en la Recuperación de Información"*, aceptado en el XXVI Congreso Argentino de Ciencias de la Computación, organizado por la Universidad Nacional de La Matanza, del 5 al 9 de octubre de 2020.

Lic. Patricia Pesado
Coordinadora Titular de RedUNCI

Mg. Jorge Eterovic
Decano DIIT UNLaM

DIIT

Universidad Nacional de La Matanza

50 UTN ANIVERSARIO FACULTAD REGIONAL SAN FRANCISCO

SAN FRANCISCO - CÓRDOBA - ARGENTINA

VIRTUAL 2020 CONAIISI

8VO CONGRESO NACIONAL

INGENIERÍA INFORMÁTICA / SISTEMAS DE INFORMACIÓN

05 06 NOV.

CERTIFICADO

DE ASISTENCIA

Por cuanto, **Ledesma Viviana Alejandra** ***D.N.I. 22.285.634*** *ha participado del 8º* ***Congreso Nacional de Ingeniería Informática / Sistemas de Información (CoNaIISI 2020)*** *organizado por la Red de Carreras de Ingeniería Informática / Sistemas de Información* ***(RIISIC)*** *perteneciente al CONFEDI, realizado de forma Virtual por la Universidad Tecnológica Nacional Facultad Regional San Francisco, los días 05 y 06 de Noviembre de 2020, se otorga el presente certificado.*

Mg.Ing. JUAN CARLOS CALLONI
Coordinador RIISIC 2020

Firma Digital

Aprobación del Documento por Juan Carlos Calloni
UNIVERSIDAD TECNOLÓGICA NACIONAL FR SAN FRANCISCO

Ing. Alberto R. TOLOZA
Decano

Firma Digital

Aprobación del Documento por Alberto Toloza
UNIVERSIDAD TECNOLÓGICA NACIONAL - FR SAN FRANCISCO

Red de Ingenierías en Informática / Sistemas de Información RIISIC CONFEDI

confedi
Consejo Federal de Decanos de Ingeniería República Argentina

Jornada

Comunicación Científica: Investigar y Publicar

Se certifica que

Viviana Ledesma

ha asistido a la Jornada "*Comunicación Científica: Investigar y Publicar"*, organizada conjuntamente por la Secretaría de Ciencia y Tecnología y la Asociación de Docentes de la UNLaM.

San Justo, martes 25 de junio de 2019.

Universidad Nacional de La Matanza
Secretaría de Ciencia y Tecnología

Mg. Ana Bidiña
Secretaria de Ciencia y Tecnología
UNLaM

AMAZON WEB SERVICES

San Justo, Febrero de 2021

Se certifica que

Fabio Quintana

DNI: 33.676.620

asistió al curso *"Competencias y Desarrollo del Talento en la Nube para la Comunidad de la UNLaM"*, dictado por docentes del DIIT certificados por Amazon Web Services, con una duración total de 30hs.

Dra. Bettina Donadello
Secretaria de Investigaciones

Mg. Jorge Eterovic
Decano DIIT

Observatorio Social
aws
DIIT
Universidad Nacional de La Matanza

elDial.cursos

Editorial Albrematica S.A., otorga el presente certificado a

VIVIANA ALEJANDRA LEDESMA

en reconocimiento por su participación en el seminario:

INTELIGENCIA ARTIFICIAL Y DERECHO

Disertado por:

Dr. Juan Corvalán - Dra. Patricia Reyes - Dr. Horacio R. Granero
Dr. Santiago Eraso Lomaquiz - Dra. Cecilia C. Danesi - Dr. Lorenzo Cotino Hueso
Dra. Susana Eloísa Mender Bini - Dr. Carlos Muñiz - Dr. Claudio Grosso

Dra. Romina Lozano
Directora Editorial

Ciudad Autónoma de Buenos Aires, 21 de diciembre de 2020

elDial.com
Contenidos Jurídicos

CONAIISI

VII Congreso Nacional de Ingeniería Informática - Sistemas de Información

2019

San Justo, 5 de diciembre de 2019

Se certifica que **Julio Bossero** ha participado como Chair en la Categoría *Bases de Datos, Artículos de investigación,* en el VII Congreso Nacional de Ingeniería Informática – Sistemas de Información, CONAIISI 2019, realizado los días 14 y 15 de noviembre en la Universidad Nacional de La Matanza.

Ing. Claudio D'Amico
Coord. Gral. CONAIISI

Dr. Carlos Neil
Coordinador RIISIC

Mg. Jorge Eterovic
Decano DIIT

UNLaM confedi DIIT RIISIC

CONAIISI

VII Congreso Nacional de Ingeniería Informática - Sistemas de Información

2019

San Justo, 5 de diciembre de 2019

Se certifica que **Hugo Ryckeboer** ha participado como Chair en la Categoría *Bases de Datos, Artículos de investigación,* en el VII Congreso Nacional de Ingeniería Informática – Sistemas de Información, CONAIISI 2019, realizado los días 14 y 15 de noviembre en la Universidad Nacional de La Matanza.

Ing. Claudio D'Amico
Coord. Gral. CONAIISI

Dr. Carlos Neil
Coordinador RIISIC

Mg. Jorge Eterovic
Decano DIIT

UNLaM confedi DIIT RIISIC

CONAIISI

VII Congreso Nacional de Ingeniería Informática - Sistemas de Información

2019

San Justo, 5 de diciembre de 2019

Se certifica que **Gabriela Cora** ha participado como Session Chair en la Categoría *Bases de Datos, Artículos de investigación,* en el VII Congreso Nacional de Ingeniería Informática – Sistemas de Información, CONAIISI 2019, realizado los días 14 y 15 de noviembre en la Universidad Nacional de La Matanza.

Ing. Claudio D'Amico
Coord. Gral. CONAIISI

Dr. Carlos Neil
Coordinador RIISIC

Mg. Jorge Eterovic
Decano DIIT

UNLaM

CONAIISI

VII Congreso Nacional de Ingeniería Informática - Sistemas de Información

2019

San Justo, 5 de diciembre de 2019

Se certifica que **Viviana Ledesma** ha participado como Session Chair en la Categoría *Bases de Datos, Artículos de investigación,* en el VII Congreso Nacional de Ingeniería Informática – Sistemas de Información, CONAIISI 2019, realizado los días 14 y 15 de noviembre en la Universidad Nacional de La Matanza.

Ing. Claudio D'Amico
Coord. Gral. CONAIISI

Dr. Carlos Neil
Coordinador RIISIC

Mg. Jorge Eterovic
Decano DIIT

UNLaM

ANEXO II

ANEXO II

EVALUACIÓN DE ALUMNOS

FORMULARIO DE EVALUACIÓN DE ALUMNOS INTEGRANTES DE EQUIPOS DE INVESTIGACIÓN

Unidad Académica: **Departamento de Ingeniería e Investigaciones Tecnológicas**
Código: **C225**
Título del Proyecto: **Resolución Eficiente de la Descomposición en Valores Singulares en una Arquitectura Híbrida y su Posterior Inserción en un Sistema de Recuperación de Información**
Director del Proyecto: Ryckeboer, Hugo Emilio
Fecha de inicio: **1/1/2019**
Fecha de finalización**: 30/04/2021**

---

1. Datos del alumno
Apellido y Nombre: **Gastón Procopio**
DNI: 35.945.222
Unidad Académica: **Departamento de Ingeniería e Investigaciones Tecnológicas**
Carrera que cursa: **Ingeniería en Informática**
Período evaluado: **1/1/2020 a 30/04/2021**

2. Dictamen de evaluación de desempeño del alumno:
Colocar una cruz donde corresponda

2.1 Satisfactorio: X
2.1 No satisfactorio:

Fundamentos del dictamen:
El alumno tuvo un desempeño satisfactorio. Mostró interés en el trabajo del equipo, aportando ideas que sumaron a la realización de las distintas implementaciones realizadas. Cumplió en tiempo y forma las tareas que le fueron asignadas.

San Justo, 15 de abril de 2021		Ryckeboer, Hugo Emilio
.....	.....	.....
Lugar y fecha	Firma del Director	Aclaración de firma

FORMULARIO DE EVALUACIÓN DE ALUMNOS INTEGRANTES DE EQUIPOS DE INVESTIGACIÓN

Unidad Académica: **Departamento de Ingeniería e Investigaciones Tecnológicas**
Código: **C225**
Título del Proyecto: **Resolución Eficiente de la Descomposición en Valores Singulares en una Arquitectura Híbrida y su Posterior Inserción en un Sistema de Recuperación de Información**
Director del Proyecto: Ryckeboer, Hugo Emilio
Fecha de inicio: **1/1/2019**
Fecha de finalización**: 30/04/2021**

---

1. Datos del alumno
Apellido y Nombre: **Fabio Quintana**
DNI: **33.676.620**
Unidad Académica: **Departamento de Ingeniería e Investigaciones Tecnológicas**
Carrera que cursa: **Ingeniería en Informática**
Período evaluado: **1/1/2020 a 30/04/2021**

2. Dictamen de evaluación de desempeño del alumno:
Colocar una cruz donde corresponda

2.1 Satisfactorio: X
2.1 No satisfactorio:

Fundamentos del dictamen:
El desempeño del alumno ha sido satisfactorio. Cumplió responsablemente las distintas tareas asignadas, respetando las consignas establecidas en el cronograma.

San Justo, 15 de abril de 2021		Ryckeboer, Hugo Emilio
.....	.....	.....
Lugar y fecha	Firma del Director	Aclaración de firma

ANEXO II

BAJAS DE INTEGRANTES

San Justo, 31 de diciembre de 2019.

Universidad Nacional de La Matanza
Departamento de Ingeniería e
Investigaciones Tecnológicas
Sr. Director del Proyecto
Ryckeboer Hugo Emilio
S _____ / _____ D

De mi mayor consideración:

Me dirijo a Usted a fin informarle mi renuncia al proyecto de investigación "Resolución Eficiente de la Descomposición en Valores Singulares en una Arquitectura Híbrida y su Posterior Inserción en un Sistema de Recuperación de Información" (código C225) a partir del 31 de diciembre del corriente año. El motivo de esta decisión es por cuestiones laborales que me dejan sin horarios disponibles para continuar con las actividades del proyecto.

Sin otro particular, la saludo muy atentamente

Julio Bossero
DNI 18.619.563

San Justo, 31 de diciembre del 2019.

Universidad Nacional de La Matanza
Departamento de Ingeniería e
Investigaciones Tecnológicas
Sr. Secretario de Investigaciones
Dra. Bettina Donadello
S _____ / _____ D

De mi mayor consideración:

Me dirijo a Usted a fin informarle la renuncia al cargo de investigador del proyecto de investigación "Resolución Eficiente de la Descomposición en Valores Singulares en una Arquitectura Híbrida y su Posterior Inserción en un Sistema de Recuperación de Información" (código C225) del docente Mg. Julio Bossero DNI 18.619.563. El motivo se debe a su falta de disponibilidad horaria por iniciar otra actividad en el ámbito profesional

Sin otro particular, saludo a Usted muy atentamente

Hugo Emilio Ryckeboer
Director del Proyecto

San Justo, 31 de diciembre de 2019.

Universidad Nacional de La Matanza
Departamento de Ingeniería e
Investigaciones Tecnológicas
Sr. Director del Proyecto
Ryckeboer Hugo Emilio
S _____ / _____ D

De mi mayor consideración:

Me dirijo a Usted a fin informarle mi renuncia al proyecto de investigación "Resolución Eficiente de la Descomposición en Valores Singulares en una Arquitectura Híbrida y su Posterior Inserción en un Sistema de Recuperación de Información" (código C225) a partir del 31 de diciembre del corriente año. El motivo de esta decisión es por cuestiones laborales que me dejan sin horarios disponibles para continuar con las actividades del proyecto.

Sin otro particular, la saludo muy atentamente

CASUSCELLI, Mauro Javier
DNI 29.661.888

San Justo, 31 de diciembre del 2019.

Universidad Nacional de La Matanza
Departamento de Ingeniería e
Investigaciones Tecnológicas
Sr. Secretario de Investigaciones
Dra. Bettina Donadello
S _____ / _____ D

De mi mayor consideración:

Me dirijo a Usted a fin informarle la renuncia al cargo de investigador del proyecto de investigación "Resolución Eficiente de la Descomposición en Valores Singulares en una Arquitectura Híbrida y su Posterior Inserción en un Sistema de Recuperación de Información" (código C225) del docente Mauro Javier Casuscelli DNI 29.661.888. El motivo se debe a su falta de disponibilidad horaria por iniciar otra actividad en el ámbito profesional

Sin otro particular, saludo a Usted muy atentamente

Hugo Emilio Ryckeboer
Director del Proyecto