Informe final publicable de proyecto
MODELAR: Modelado del desempeñO de métoDos numÉricos en pLataformas de hArdware heteRogéneas

Código de proyecto ANII: FCE_3_2022_1_172419

Fecha de cierre de proyecto: 01/11/2025

DUFRECHOU LASCA, Ernesto (Responsable Técnico - Científico)
FAVARO SAPRIZA, Federico (Co-Responsable Técnico-Científico)
FREIRE PICÓN, Manuel (Investigador)
MARICHAL CHÁVEZ, Raúl Ignacio (Investigador)
SEVESO GIORDANO, Franco (Investigador)
USLENGHI, Florencia (Investigador)
BENTURA, Mateo (Investigador)
BERGER, Gonzalo (Investigador)
FERREIRA QUAGLIATA, Maria Jimena (Investigador)

UNIVERSIDAD DE LA REPÚBLICA. FACULTAD DE INGENIERÍA (Institución Proponente) \\ UNIVERSIDAD DE LA REPÚBLICA. FACULTAD DE INGENIERÍA

Resumen del proyecto

El álgebra lineal numérica es una herramienta fundamental en muchas áreas de la ciencia y la ingeniería, como la inteligencia artificial, la optimización, la computación gráfica y el control de sistemas. Desde hace décadas, resolver este tipo de problemas requiere una gran capacidad de cómputo, lo que ha impulsado el desarrollo y el uso de tecnologías de computación de alto desempeño. Hoy en día, los sistemas de cómputo combinan distintos tipos de procesadores y aceleradores, y para un mismo problema pueden existir varias formas de resolverlo, sin que siempre sea evidente cuál es la más conveniente.

Este proyecto se centra en dos objetivos principales. El primero es desarrollar métodos que permitan resolver estos problemas de manera más eficiente, aprovechando al máximo las capacidades de los procesadores y aceleradores modernos, como los utilizados en centros de cómputo especializados.

El segundo objetivo es analizar qué recursos necesita cada método —por ejemplo, memoria, tiempo de ejecución y consumo de energía— según el tipo de problema y el hardware utilizado. Además, se busca desarrollar herramientas capaces de elegir automáticamente la mejor combinación entre el método de resolución y la plataforma de cómputo en cada situación concreta, ya sea mediante modelos matemáticos o técnicas de aprendizaje automático. De esta forma, se apunta a utilizar mejor los recursos disponibles y a reducir tanto el tiempo de cálculo como el consumo energético.

Ciencias Naturales y Exactas / Ciencias de la Computación e Información / Ciencias de la Computación / Computación de alto desempeño, Álgebra Lineal Numérica
Palabras clave: Álgebra lineal numérica / Modelo de predicción de desempeño / Arquitecturas de hardware heterogéneas /

Antecedentes, problema de investigación, objetivos y justificación.

La resolución de problemas de gran escala o con restricciones temporales, suelen requerir soluciones especializadas de hardware y software, categorizadas bajo el área de computación de alto desempeño (HPC por su sigla en inglés). Desde el punto de vista del software, una estrategia ampliamente utilizada para obtener un buen desempeño consiste en expresar los problemas en términos de operaciones matriciales, lo que permite aprovechar bibliotecas con rutinas de álgebra lineal numérica altamente perfeccionadas y diseñadas para obtener el mejor desempeño de cada plataforma de hardware. Es por esto que el ALN atraviesa diversas áreas de la ciencia y la ingeniería, como inteligencia artificial, reconocimiento de patrones, optimización, computación gráfica y control [29].
Por su parte, las plataformas de HPC han tendido a integrar múltiples procesadores, con el fin de aumentar el desempeño a través del procesamiento paralelo de distintas tareas o datos. Esto se ha acentuado especialmente en las últimas dos décadas, debido a la desaceleración en el aumento de la densidad de transistores alcanzable en una pastilla de silicio [31]. Como consecuencia, la totalidad de las CPUs actuales ofrecen la posibilidad de ejecutar tareas en paralelo sobre varios núcleos y, para problemas de dominio específico, es común el uso de plataformas híbridas, que complementan el trabajo de las CPU convencionales con procesadores altamente paralelos, comúnmente denominados aceleradores de hardware.
Un ejemplo paradigmático de este tipo de dispositivos son los procesadores gráficos (GPUs), diseñados originalmente para liberar a la CPU de los cálculos necesarios para presentar gráficos en la pantalla. Debido a su impresionante rendimiento, precios razonables y una atractiva relación entre capacidad de cómputo y consumo energético, las GPUs han sido adoptadas para realizar cómputos de propósito general, volviéndose un componente clave en áreas como HPC o Inteligencia Artificial [34].
En forma más reciente, la masiva adopción de las GPUs ha motivado que algunos fabricantes de procesadores tradicionales ingresen en el mercado de los aceleradores de hardware presentando propuestas alternativas. En este sentido, se destacan versiones modernas de las Field Programmable Gate Arrays (FPGAs), Digital Signal Processors (DSPs) y los Tensor Processing Unit (TPUs), estos últimos desarrollados por la empresa Google para el campo de la inteligencia artificial (ofrecidos como aceleradores en dispositivos autónomos de bajo consumo [2]).
El aumento del desempeño mediante la integración de una gran cantidad de procesadores, donde cada uno de ellos consume energía sea utilizado o no, ha despertado la preocupación por la eficiencia energética, tanto del hardware como del software. A manera de ejemplo, la continua expansión y modernización de las plataformas de HPC en grandes centros de cómputo permite prever el alcance de sistemas de Exaescala dentro de poco tiempo [7] (de hecho, desde noviembre de 2021 la supercomputadora japonesa Fugaku ya es capaz de alcanzar 1 Exaflop/s en simple precisión y precisiones inferiores [1]). Sin embargo, con las tasas actuales de desempeño por energía consumida, el crecimiento del poder de cómputo de estos sistemas es acompañado por serios desafíos económicos y ambientales. Es importante subrayar que el costo asociado al consumo energético de estas plataformas de hardware puede ser incluso superior al costo de la adquisición de la propia plataforma. El problema de la eficiencia energética en las arquitecturas de hardware se postula por lo tanto, como una limitante que debe ser superada para el desarrollo tecnológico a nivel global, y en especial para países como el nuestro, donde los recursos financieros para mantener estas plataformas de cómputo son escasos.
A nivel global, un ejemplo claro de esta preocupación es la lista Green500 (www.green500.org), que ordena la lista Top500, de las 500 supercomputadoras más potentes del mundo publicada en forma semestral, en base a su eficiencia energética medida en FLOPS/Watt. Por otro lado, existen contextos en los que el consumo de energía presenta restricciones incluso más severas, especialmente considerando el impresionante desarrollo de las aplicaciones en los temas de IoT (Internet of Things) y Smart Cities. Algunos ejemplos pueden ser los dispositivos celulares, vehículos aéreos no tripulados (UAVs), sensores autónomos, o wereables. Los usuarios de estos dispositivos cada día demandan mayores niveles de cómputo (procesamiento de imágenes, algoritmos de optimización, etc.) y un menor consumo energético para satisfacer los requerimientos de autonomía. En este sentido, aumentar la eficiencia en el uso de los dispositivos permite, en muchos casos, además de alcanzar ahorros económicos

viabilizar la implementación de soluciones. Es así que en los últimos años han surgido distintos esfuerzos orientados al desarrollo de procesadores de bajo consumo energético y a su utilización para resolver operaciones de ALN [3,38,40,43].
La realidad descrita anteriormente muestra una tendencia en la que es cada vez más común que las plataformas de hardware integren distintos tipos de procesadores con capacidades dispares, de forma de atender requerimientos particulares de cómputo. Desde el diseño software, esta tendencia implica también desafíos, que pasan por el re-diseño de distintas aplicaciones para adaptarse a las plataformas de hardware disponibles, mejorando así su velocidad de cómputo y consumo energético. En este contexto, es necesario caracterizar el uso de recursos y el desempeño que alcanzaría una aplicación, trabajando sobre determinada entrada de datos, en cada una de las opciones de hardware. De esta forma se puede escoger dinámicamente, y dependiendo de las restricciones impuestas (por ejemplo de tiempo de ejecución o consumo energético) qué método y plataforma emplear.
Si bien la idea de que las rutinas de ALN sean capaces de adaptarse a distintas plataformas de hardware y entradas de datos se ve acentuada en el presente, el modelado del comportamiento y el desarrollo de kernels adaptativos tiene larga data, remontándose a los trabajos pioneros en esta área. Como ejemplos destacados en el campo del álgebra sobre matrices densas figuran el desarrollo de la biblioteca ATLAS [14], una implementación del estándar BLAS que se autoajustaba a las características del hardware donde ejecutaba, y los esfuerzos de Demmel y Volkov [47] por modelar el comportamiento de la multiplicación de matrices en los comienzos del uso de las GPUs para problemas de propósito general.
Para el caso del álgebra sobre matrices dispersas también se han desarrollado una gran cantidad de trabajos con foco en el modelado del desempeño de kernels, a veces complementado con la implementación de rutinas adaptativas. Es importante mencionar que la irregularidad de las estructuras de datos en problemas dispersos agrega una dimensión de dificultad más al modelado del desempeño, ya que el desempeño de los kernels suele variar enormemente según la disposición de los elementos no nulos en las matrices de entrada, y no solo según el tamaño de la entrada como en problemas densos. Esto genera que distintos métodos para resolver la misma operación obtengan desempeños disímiles según las características de la entrada, y que sea difícil determinar cuál es la mejor rutina para cierta operación.
En el último período algunos de los principales trabajos sobre álgebra dispersa se centran en diseñar estructuras de datos para almacenar las matrices dispersas que maximicen el uso del ancho de banda de memoria en los distintos dispositivos [9], balancear la carga de los distintos hilos [36][15][8], mejorar el despacho de hilos según las dependencias de datos [37][18], entre otras cosas. Esto ha dado lugar a una considerable cantidad de rutinas que muestran fortalezas sobre distintos procesadores, bajo ciertas entradas de datos, pero sobre las cuales es difícil establecer reglas generales que caractericen estas fortalezas.
Esfuerzos recientes han apuntado a un modelado del desempeño que permita seleccionar automáticamente entre distintas implementaciones de cierta rutina [22][6][20][19], entre distintas estructuras de datos para representar las matrices dispersas [44][50][39], entre distintas precisiones numéricas [30], u otros aspectos del algoritmo [32][49]. En forma más general y con foco en el modelado de desempeño, tanto de álgebra densa como dispersa, es necesario destacar la propuesta del roofline model, presentado por el grupo de investigación de D. Patterson [48], y diversas extensiones de esta técnica [33][45][10].
Este proyecto plantea investigar el modelado del desempeño de algunos kernels clave de ALN densa y dispersa, desde el punto de vista del tiempo de ejecución y el consumo de recursos en distintos dispositivos. Tomando como base los esfuerzos realizados por investigadores del equipo, así como los principales resultados del estado del arte, se propone mejorar las técnicas existentes de predicción y modelado del desempeño, basadas principalmente en modelos de aprendizaje automático de caja negra. Se prevé que la profundización en esta línea, permita pasar de la aplicación de modelos de caja negra a modelos explicables, que aporten conocimientos sobre los métodos. Se avanzará hacia el desarrollo de modelos analíticos basados en Roofline Model (RLM) que integren información sobre métodos y plataformas. Los avances en estas temáticas permitirían el desarrollo de bibliotecas adaptativas, multiplataforma, capaces de seleccionar las mejores rutinas según el contexto y así mejorar drásticamente el desempeño de aplicaciones científicas.

El objetivo general del proyecto es caracterizar el desempeño de un conjunto de rutinas clave que representen el estado del arte del ALN densa y dispersa sobre distintas plataformas de hardware paralelo. En particular, se apunta a desarrollar modelos que permitan predecir cuál es la rutina más adecuada para resolver cierta operación, sobre determinados datos, bajo ciertas restricciones de tiempo de ejecución o consumo energético. La predicción utilizando dichos modelos, además de ser precisa, debe tener un bajo costo computacional, ya que el costo de realizar la predicción no debe superar el potencial ahorro al ejecutar la mejor rutina en cada caso.

Entre los objetivos específicos se encuentra la actualización del laboratorio de medición de desempeño y energía, definir y organizar un conjunto de rutinas de ALN a estudiar, actualizar las rutinas desarrolladas por el grupo de investigación, adaptándolas a las nuevas generaciones de hardware, evaluar las técnicas analíticas y basadas en aprendizaje automático para el modelado del desempeño, diseñar modelos capaces de predecir el desempeño de rutinas para distintos problemas, identificar posibles mejoras a partir de los modelos, y avanzar en la formación de posgrado de los integrantes del equipo.

Metodología/Diseño del estudio

Se elaboraron modelos para predecir el desempeño de métodos de ALN. Para avanzar en este objetivo, se actualizó el laboratorio de medición de energía [16] y se definió un conjunto de rutinas y plataformas para su estudio. Por este motivo, el proyecto se dividió en dos grandes etapas: una preliminar, en la que se actualizaron las plataformas de experimentación y se definió el conjunto de rutinas y plataformas, y otra de modelado, en la que se desarrollaron modelos capaces de predecir el desempeño del software y del hardware definidos en la etapa previa. De forma transversal a estas etapas, se siguió un enfoque de trabajo en equipo, buscando una división de tareas que permitió avanzar en paralelo sobre distintas sublíneas (es decir, modelos analíticos y basados en aprendizaje automático). Si bien el diseño, la implementación y la evaluación de software, así como el desarrollo de modelos analíticos, se realizaron de forma coordinada, el trabajo en cada plataforma fue a cargo de un subequipo. El avance global fue coordinado por los responsables, quienes se encargaron de definir, junto con los referentes de cada subequipo, las estrategias generales y las tareas concretas de cada línea de trabajo.

Etapa preliminar.

i) Actualización del laboratorio de medición de energía.

Para facilitar la evaluación energética, se implementó una infraestructura que permitió la interconexión sencilla entre plataformas y dispositivos de medición, posibilitando el acceso externo. En esta etapa, se adaptó el laboratorio a las nuevas plataformas de hardware estudiadas en el proyecto y a las necesidades surgidas en las etapas iniciales. También se adquirió nuevo hardware y se extendió el laboratorio hacia el final del proyecto.

ii) Definición del conjunto de rutinas y plataformas a estudiar.

Para desarrollar modelos eficaces de las principales operaciones de ALN en distintas plataformas, se seleccionaron rutinas y plataformas de hardware representativas del estado del arte. Con este fin, se actualizó el estado del arte de las plataformas consideradas y de los métodos computacionales disponibles para cada una. Esto permitió disponer de un conocimiento actualizado y detallado de las arquitecturas de hardware, identificar los aspectos en los que era posible aportar conocimiento original y prever las principales dificultades a enfrentar.

Además de realizar una evaluación detallada de las principales rutinas públicamente disponibles para las operaciones de ALN seleccionadas, se actualizaron las desarrolladas en el seno de nuestro grupo de investigación, refinando su desempeño y adaptándolas a nuevas generaciones de hardware. Asimismo, se desarrollaron variantes de los métodos del grupo para plataformas que no habían sido abordadas previamente.

El desarrollo y mejoramiento de estas rutinas siguieron una estrategia iterativa-incremental basada en prototipos [13]. Esta consistió en el diseño y elaboración rápida de prototipos enfocados en subconjuntos de objetivos concretos, que permitieron validar las estrategias definidas mediante experimentación en etapas tempranas. De esta manera se mitigaron riesgos asociados a la dedicación de tiempo y esfuerzo a estrategias basadas en hipótesis erróneas. Cada prototipo fue extendido y mejorado a través de distintas iteraciones que incluyeron subetapas de diseño, implementación y validación experimental. Una vez alcanzados los objetivos planteados, se procedió a difundir los resultados obtenidos.

Para la evaluación experimental, en el caso de las rutinas sobre matrices densas, se generaron conjuntos de matrices sintéticas de distintos tamaños. En el caso de matrices dispersas, dado que el patrón de elementos no nulos incide fuertemente en el desempeño, se utilizaron matrices provenientes de problemas reales de la Suite Sparse Matrix Collection (https://sparse.tamu.edu).

Etapa de desarrollo de modelos de predicción.

Se trabajó para avanzar en forma simultánea en las dos sublíneas. Como primer paso, se actualizó el estado del arte sobre el desarrollo de modelos de predicción en ambos paradigmas. Para los modelos analíticos, se partió de modelos de alto nivel de abstracción y, de forma gradual, se incorporaron características específicas de los dispositivos estudiados (memorias caché, unidades funcionales, entre otras). Se utilizaron variantes del Roofline Model como línea base. La evaluación y validación de los desarrollos se realizaron utilizando diferentes dispositivos abordados y empleando las matrices descritas previamente.

En cuanto a los modelos basados en aprendizaje automático, se determinaron los métodos y las features a considerar. En una etapa preliminar se evaluó una amplia gama de clasificadores de “caja negra”, incluyendo SVM, KNN y distintos clasificadores de ensemble. Los resultados obtenidos mostraron que algunos clasificadores de ensemble alcanzaron desempeños alentadores. Se identificó que la elección de features constituye uno de los puntos críticos para obtener resultados precisos, dado que su cómputo puede implicar costos comparables o incluso superiores al del problema a resolver.

El estudio y validación requiere conjuntos extensos de matrices. Para conjuntos de datos reducidos se utilizó la técnica de validación cruzada, que particiona el conjunto de datos y realiza la validación en múltiples iteraciones, alternando los conjuntos de entrenamiento y de predicción entre las distintas particiones. Para disponer de conjuntos de datos más amplios destinados al entrenamiento y la predicción, se generaron matrices sintéticas.

En el caso denso, los parámetros relevantes son la cantidad de filas y columnas de las matrices, lo que permite generar conjuntos sintéticos sin mayores dificultades. En contraste, para matrices dispersas, la cantidad y disposición de los elementos distintos de cero impacta directamente en los patrones de acceso a memoria, por lo que la generación de conjuntos efectivos no resulta trivial. Dado que generar todos los patrones posibles de no-ceros es inviable, se optó por reproducir y parametrizar patrones de dispersión derivados de problemas reales. Se exploró esta línea utilizando técnicas de IA generativa para ampliar el conjunto disponible de matrices dispersas triangulares de la colección SuiteSparse.

En paralelo con esta etapa se exploró la aplicación de técnicas basadas en programación genética provenientes de la ingeniería química computacional.

Resultados, análisis y discusión

El proyecto tuvo como objetivo general avanzar en el desarrollo de métodos, herramientas y modelos para comprender, predecir y optimizar el

desempeño de rutinas de álgebra lineal numérica (ALN) en arquitecturas de cómputo modernas, con especial énfasis en plataformas paralelas y energéticamente eficientes. A lo largo de su ejecución se obtuvieron resultados relevantes en tres líneas principales: desarrollo y mejoramiento de rutinas numéricas, construcción de modelos analíticos de desempeño y desarrollo de modelos basados en aprendizaje automático para selección y optimización automática de algoritmos. De forma complementaria, el proyecto contribuyó significativamente a la formación de estudiantes de grado y posgrado mediante proyectos de grado, tesis y trabajos de investigación.

En la línea de desarrollo y mejoramiento de rutinas, uno de los principales resultados estuvo vinculado a la resolución de sistemas triangulares dispersos (Sparse Triangular Solve, SpTRSV) en GPUs, operación fundamental en numerosos métodos iterativos y precondicionadores utilizados en computación científica. Dado que el desempeño de estos métodos depende fuertemente de la explotación eficiente del paralelismo disponible, se propuso un nuevo enfoque de análisis por niveles (level-set analysis) orientado a mejorar la planificación de la ejecución paralela durante la fase de resolución. La estrategia desarrollada modificó la construcción de los niveles para adelantar la ejecución de filas con dependencias críticas. Como complemento, se diseñó un nuevo formato interno de almacenamiento de matrices dispersas orientado a acelerar la fase de resolución mediante accesos a memoria más eficientes. Este diseño implicó un incremento moderado en los requerimientos de memoria, pero permitió reducir significativamente el tiempo total de ejecución. La evaluación experimental mostró mejoras de desempeño de hasta un 70 % respecto a implementaciones recientes basadas en estrategias synchronization-free, superando además a diversos solucionadores considerados estado del arte, especialmente en escenarios donde múltiples sistemas lineales deben resolverse utilizando la misma matriz [51].

En estrecha relación con este resultado, se abordó la paralelización de la factorización incompleta LU (ILU), ampliamente utilizada como precondicionador algebraico para acelerar la convergencia de métodos iterativos. Si bien los solucionadores triangulares synchronization-free han sido extensamente estudiados, la aplicación sistemática de este paradigma a la factorización ILU había recibido menor atención en la literatura. En este contexto se desarrollaron implementaciones synchronization-free del precondicionador ILU-0 para GPUs. Se propusieron tres variantes del algoritmo que difieren en la forma en que se realizan las actualizaciones de filas luego de cada eliminación de coeficientes, así como una cuarta variante que incorpora un análisis previo por niveles para optimizar el orden de ejecución [52].

En la línea de modelos analíticos, el proyecto avanzó en el uso del Roofline Model (RLM) como herramienta para comprender y guiar la optimización de kernels de ALN en hardware heterogéneo. En este contexto se propuso una extensión del RLM capaz de considerar simultáneamente métricas de tiempo de ejecución y consumo energético para kernels de álgebra lineal dispersa que trabajan sobre formatos a bloques. Guiados por el modelo analítico desarrollado, se desarrollaron implementaciones de la multiplicación matriz dispersa-vector (SpMV) para FPGAs que utilizan formatos de almacenamiento disperso a bloques para lograr un mejor patrón de acceso a la memoria [53].

Adicionalmente, se realizaron avances orientados a la reducción de comunicaciones en kernels dispersos, abordando uno de los principales cuellos de botella en arquitecturas paralelas modernas. Estos estudios contribuyeron a caracterizar el impacto relativo del movimiento de datos frente al cómputo. Concretamente, se realizó una evaluación del impacto de distintas técnicas de reordenamiento y compresión de índices en el espacio necesario para almacenar las matrices de la colección SuiteSparse. Los resultados mostraron importantes reducciones, que pueden desembocar en mejoras de desempeño en las rutinas de ALN al reducir la cantidad de datos que se transfieren desde la memoria a los procesadores [54].

La tercera gran línea de resultados correspondió al desarrollo de modelos basados en aprendizaje automático para optimizar decisiones algorítmicas que tradicionalmente requieren intervención manual experta. Debido a que el desempeño de kernels dispersos depende fuertemente del patrón de dispersión de las matrices, seleccionar la representación o el algoritmo adecuado constituye un problema complejo y altamente dependiente de los datos.

Se desarrollaron varios estudios preliminares orientados a la selección automática de implementaciones de SpMV mediante aprendizaje automático. En estos estudios se construyeron modelos capaces de predecir, a partir de características numéricas de las matrices (como dimensiones, densidad y métricas estructurales) cuál de varias implementaciones disponibles resultaría más eficiente. Se evaluaron enfoques supervisados y no supervisados, identificándose desafíos asociados al desbalance de clases dentro de los conjuntos de entrenamiento. A pesar de estas dificultades, modelos basados en Random Forest alcanzaron precisiones cercanas al 80% en la clasificación entre cinco métodos alternativos.

Un resultado particularmente relevante fue la incorporación de características visuales derivadas de representaciones estructurales de matrices dispersas. Mediante la representación del patrón de no ceros de las matrices como un mapa de bits y la extracción de descriptores visuales --como el histograma de gradientes (HOG) y SIFT-- se logró capturar información estructural difícil de representar mediante métricas numéricas tradicionales. La combinación de macro-bloques 4×4, reducción dimensional mediante SVD truncado y clasificadores Random Forest alcanzó un valor Macro-F1 de 0.8909, superando ampliamente a modelos basados exclusivamente en features numéricas. El análisis del frente de Pareto entre precisión y latencia permitió identificar configuraciones que equilibran calidad predictiva y costo computacional. En particular, combinaciones basadas en Gradient Boosting, con mapas de bits de 32×32 y reducción SVD bidimensional mostraron un compromiso favorable para aplicaciones sensibles al tiempo, manteniendo alta precisión con un costo adicional prácticamente marginal. Los experimentos confirmaron que el tiempo requerido para la extracción de características visuales resulta despreciable frente a las mejoras obtenidas, lo que valida su aplicabilidad en escenarios reales.

Otro resultado destacado del proyecto fue el desarrollo de un sistema de selección automática de algoritmos para SpTRSV mediante aprendizaje automático, realizado en el marco de un proyecto de grado [55]. Se evaluaron 17 modelos pertenecientes a distintas familias --incluyendo métodos de ensemble, boosting, modelos lineales y redes neuronales-- utilizando matrices de la colección SuiteSparse. Para evaluar adecuadamente el

impacto práctico de las predicciones se definió una métrica denominada TAR, que cuantifica el sobrecosto temporal respecto a la elección óptima del algoritmo.

El estudio incluyó además el diseño de nuevas características estructurales de matrices, entre ellas una distribución por bandas horizontales, el índice de Gini aplicado a la dispersión y el denominado P-ratio. Asimismo, se desarrolló una técnica de etiquetado capaz de identificar algoritmos con comportamientos equivalentes, evitando penalizar predicciones indistinguibles desde el punto de vista del desempeño.

Con el objetivo de ampliar el conjunto de entrenamiento se implementó un mecanismo de generación de matrices sintéticas mediante Redes Generativas Adversariales (GAN). Las matrices se representaron como imágenes de 128×128 píxeles que codifican la densidad de elementos no nulos en cada región, permitiendo entrenar un modelo generativo capaz de producir nuevas instancias estadísticamente válidas. Este proceso permite generar matrices sintéticas adicionales, ampliando significativamente la diversidad de los conjuntos de datos para el entrenamiento de los métodos de aprendizaje automático y mejorando los resultados de los predictores sobre instancias reales nunca vistas.

Utilizando modelos de Gradient Boosting entrenado sobre un dataset de 3500 matrices generadas mediante la GAN, la selección automática de algoritmos introdujo un sobrecosto promedio cercano al 4% respecto al tiempo óptimo, en contraste con el 872% esperado al seleccionar algoritmos de forma aleatoria. Estos resultados evidencian el potencial del enfoque para escenarios reales donde múltiples sistemas deben resolverse sobre una misma matriz, amortizando el costo del preprocesamiento y la predicción.

Actualmente se está trabajando para consolidar estos resultados y validarlos en la comunidad mediante publicaciones arbitradas.

Además de los resultados científicos y tecnológicos, el proyecto tuvo un impacto significativo en la formación de recursos humanos. Diversos estudiantes participaron activamente en el desarrollo experimental, implementación de algoritmos y construcción de modelos predictivos, integrando actividades de investigación con proyectos de grado y tesis. En particular, durante el transcurso del proyecto se desarrolló un proyecto de fin de carrera estrechamente relacionado con la temática del proyecto, culminaron su maestría Manuel Freire [56] y Gonzalo Berger [57] (integrantes del equipo) y culminó su doctorado Federico Favaro [58] (co-responsable del proyecto). Otros integrantes del equipo lograron importantes avances en sus posgrados.

Los resultados principales del proyecto, junto con contenidos más orientados a generar interés en temáticas afines, se divulgaron al público general mediante dos piezas audiovisuales (un video largo y uno corto). Las piezas se difundieron a partir de febrero de 2026 en Instagram y LinkedIn. Los alcances de los videos fueron los siguientes:
En Instagram
Video largo: https://www.instagram.com/p/DUl0_5zETao/?utm_source=ig_web_copy_link&igsh=MzRlODBiNWFlZA==
18.000 visualizaciones
Video corto: https://www.instagram.com/reel/DUjFLDsjK7z/?utm_source=ig_web_copy_link&igsh=MzRlODBiNWFlZA==
11.000 visualizaciones

En Linkedin
https://www.linkedin.com/posts/fingudelar_modelar-divulgaciaejncientaedfica-computaciaejncientaedfica-activity-7427084674054811648-yLh9?utm_source=social_share_send&utm_medium=member_desktop_web&rcm=ACoAADeFE2IBxLJ_OOg6tes1-PTwYcTQ_QF_RDU
4.000 visualizaciones de la publicación
1.500 reproducciones del video

Los dos videos se subieron al repositorio Colibrí y los links públicos son incluidos dentro de los resultados del proyecto.

Conclusiones y recomendaciones

El proyecto tuvo como objetivo central avanzar en la comprensión, modelado y optimización del desempeño de métodos de álgebra lineal numérica (ALN) en arquitecturas de cómputo modernas, abordando simultáneamente tres dimensiones complementarias: el desarrollo de nuevas rutinas numéricas eficientes, la construcción de modelos analíticos capaces de explicar el desempeño observado y la incorporación de técnicas de aprendizaje automático para automatizar decisiones algorítmicas dependientes de los datos. A partir de los resultados obtenidos puede afirmarse que el proyecto cumplió satisfactoriamente sus objetivos generales y permitió consolidar esta línea de investigación.

Una primera conclusión es que la combinación de enfoques tradicionales de optimización algorítmica con herramientas modernas de modelado y aprendizaje automático puede conducir a importantes ahorros de tiempo y energía. La experiencia del programador, así como su conocimiento del problema y del hardware subyacente, continúa siendo clave en el desarrollo de kernels de ALN de alto desempeño. Sin embargo, los problemas de ALN dispersa presentan desafíos extremadamente complejos relativos al acceso a memoria y localidad de los datos, lo cual dificulta el desarrollo de rutinas que sean capaces de explotar al máximo el potencial del hardware en todas las situaciones. Los resultados del proyecto muestran que es viable una metodología en la que se desarrollen varias rutinas de ALN para una misma operación, cada una diseñada para obtener su mejor rendimiento bajo ciertas hipótesis, y luego se delegue en un método de aprendizaje automático la elección de la rutina más adecuada para cada situación.

En la línea de desarrollo de rutinas, los resultados obtenidos en la resolución de sistemas triangulares dispersos y en la factorización incompleta

ILU confirmaron que aún existen oportunidades significativas de mejora incluso en problemas ampliamente estudiados. Las modificaciones introducidas en el análisis por niveles y el diseño de formatos de almacenamiento demostraron que pequeñas variaciones en la organización de dependencias y accesos a memoria pueden producir mejoras sustanciales de desempeño en GPUs modernas. En el caso de la ILU se da un ejemplo paradigmático del enfoque metodológico descrito anteriormente. El trabajo sobre esta operación desembocó en un conjunto de rutinas donde, si bien hay algunas que superan a otras en general, cada una de ellas sobresale en casos puntuales y difíciles de caracterizar. Además, esta operación es relativamente costosa (en particular si se admite fill-in durante la factorización), y suele ejecutarse una única vez por matriz (a diferencia del solver triangular). Esto vuelve especialmente relevante la existencia de un mecanismo que sea capaz de elegir la rutina más adecuada para cada matriz mediante un modelo liviano desde el punto de vista del tiempo de ejecución.

En relación con los modelos analíticos, el proyecto confirmó la utilidad del Roofline Model como herramienta no solo descriptiva sino también prescriptiva. La extensión desarrollada permitió incorporar consideraciones energéticas junto con métricas de desempeño, lo cual resulta especialmente relevante en el contexto actual, donde la eficiencia energética se ha convertido en un criterio de diseño tan importante como el tiempo de ejecución.
La experiencia adquirida mostró que, si bien el RLM es un modelo sencillo, es flexible y útil para guiar el diseño de rutinas de ALN. Además, puede utilizarse para seleccionar la mejor opción entre distintas rutinas, mejorando el tiempo de ejecución y consumo energético, prácticamente sin ningún esfuerzo computacional adicional. Por último, si bien la extensión del modelo fue evaluada usando la operación SpMV en FPGAs, esta es parametrizable y puede utilizarse directamente sobre otras operaciones y plataformas.

Si bien los modelos analíticos tienen la ventaja de ser interpretables y generalmente rápidos de aplicar, también se observó que presentan limitaciones cuando deben capturar comportamientos altamente variables acorde a la estructura de los datos, como ocurre frecuentemente en kernels dispersos. Esta observación justifica la incorporación de técnicas de aprendizaje automático.

Una de las conclusiones principales en esta línea es que la representación de la información estructural de las matrices constituye un factor crítico para el éxito de los modelos predictivos, y que lograr dicha representación en un tiempo mucho menor que los potenciales ahorros de tiempo de ejecución que se pretende lograr a través de la aplicación de estas técnicas es un desafío complejo. Los experimentos confirmaron que las características numéricas tradicionales (dimensión, cantidad de no ceros, promedio de no ceros por fila, etc.) capturan solo parcialmente la complejidad de los patrones de dispersión. La incorporación de descriptores visuales derivados de representaciones tipo imagen permitió capturar propiedades globales de la estructura matricial que influyen directamente en el comportamiento de los accesos a memoria. Si bien esta línea presenta resultados alentadores, resta determinar definitivamente si el costo de aplicación de estas técnicas resulta práctico, es decir, permite un ahorro significativo en el tiempo total.

Otra conclusión es que, además de la obtención de características que capturen fielmente la información de las matrices, los modelos de aprendizaje automático para predecir el rendimiento de rutinas de ALN dispersa requieren un ajuste cuidadoso de hiperparámetros y los conjuntos de entrenamiento. En particular, el desbalance de los conjuntos de entrenamiento puede afectar significativamente el desempeño de los clasificadores. En este caso, la generación de matrices sintéticas mediante técnicas generativas es una estrategia viable para mitigar parcialmente este problema y expandir los espacios de entrenamiento cuando los datos reales son limitados. Sin embargo, la generación de conjuntos sintéticos que agreguen diversidad y no reproduzcan los sesgos del conjunto original, sobreajustándose al mismo, continúa siendo un desafío importante.

Desde el punto de vista del impacto científico y tecnológico, el proyecto contribuyó a consolidar una línea de investigación que había comenzado a desarrollar el grupo Heterogeneous Computing Lab (HCL) del Instituto de Computación (INCO). Durante el proyecto se realizaron varios trabajos de grado y posgrado, donde una significativa cantidad de estudiantes tuvo la oportunidad de trabajar en temáticas afines al proyecto, comprendiendo los desafíos científicos y tecnológicos que se plantean. En particular, varios estudiantes de posgrado, miembros del equipo de investigación, avanzaron en sus estudios y realizaron tesis sobre temáticas afines. Esto contribuye a fortalecer las capacidades locales de investigación.

En función de las conclusiones anteriores, se recomienda continuar profundizando la integración entre modelado de desempeño y técnicas de inteligencia artificial, avanzando hacia sistemas capaces de auto-adaptarse dinámicamente al hardware y a los datos de entrada. Asimismo, resulta interesante extender los estudios hacia arquitecturas emergentes, incluyendo aceleradores de IA y plataformas edge, donde las restricciones energéticas y de memoria amplifican la importancia de la optimización automática. Para esto, también sería necesario profundizar el estudio para incorporar de manera más detallada la dimensión energética en los modelos. En particular, podrían utilizarse mecanismos automáticos para seleccionar entre modos de fucionamiento del hardware que impliquen menor consumo energético cuando esto tenga un impacto mínimo (o admisible) en el desempeño.

Finalmente, el proyecto permitió validar la hipótesis de que una metodología recomendable para maximizar el desempeño del software numérico es contar con múltiples rutinas especializadas que actúen en conjunto con herramientas capaces de decidir automáticamente cuál ejecutar en función del contexto de hardware y datos, más que contar con una rutina para cada operación que obtenga el mejor desempeño en la mayoría de los casos.

Productos derivados del proyecto

Tipo de producto	Título	Autores	Identificadores	URI en repositorio de Silo	Estado
Publicación de trabajo en evento (artículo de conferencia)	A synchronization-free incomplete LU factorization for GPUs with level-set analysis	Manuel Freire; Ernesto Dufrechou; Pablo Ezzatti	https://doi.org/10.1109/PDP66500.2025.00037	https://hdl.handle.net/20.500.12008/53706	Finalizado
Artículo científico	Optimizing the Performance of the Sparse Matrix–Vector Multiplication Kernel in FPGA Guided by the Roofline Model	Federico Favaro, Ernesto Dufrechou, Juan P. Oliver, Pablo Ezzatti	https://doi.org/10.3390/mi14112030	https://hdl.handle.net/20.500.12008/53701	Finalizado
Publicación de trabajo en evento (artículo de conferencia)	Trajectory-based Metaheuristics for Improving Sparse Matrix Storage	Manuel Freire; Raúl Marichal; Ernesto Dufrechou; Pablo Ezzatti; Martín Pedemonte	https://doi.org/10.1109/LA-CCI58595.2023.10409303	https://hdl.handle.net/20.500.12008/53697	Finalizado
Publicación de trabajo en evento (artículo de conferencia)	Towards Reducing Communications in Sparse Matrix Kernels	Manuel Freire; Raúl Marichal; Ernesto Dufrechou; Pablo Ezzatti	https://doi.org/10.1007/978-3-031-40942-4_2	https://hdl.handle.net/20.500.12008/53694	Finalizado
Tesis de doctorado	Exploring Field-Programmable Gate Arrays and High-Level Synthesis to perform numerical linear algebra operations for High Performance Computing.	Federico Favaro	ISSN 1688-2784	https://hdl.handle.net/20.500.12008/52959	Finalizado

Tipo de producto	Título	Autores	Identificadores	URI en repositorio de Silo	Estado
Tesis de grado/monografías	Selección óptima de rutinas de álgebra lineal dispersa usando aprendizaje automático	Mauricio Friss de Kereki		https://hdl.handle.net/20.500.12008/53731	Finalizado
Tesis de maestría	Aplicación de estrategias Synchornization Free a la resolución de operaciones de Algebra Lineal Dispersa.	Manuel Freire		https://hdl.handle.net/20.500.12008/50195	Finalizado
Tesis de maestría	Uso de formatos de almacenamiento y algoritmos a bloques en álgebra dispersa en dispositivos masivamente paralelos.	Gonzalo Berger Álvarez		https://hdl.handle.net/20.500.12008/46179	Finalizado
Tesis de grado/monografías	Selección óptima de rutinas de álgebra lineal dispersa usando aprendizaje automático	Mauricio Friss de Kereki		https://hdl.handle.net/20.500.12008/53731	Finalizado
Material didáctico	Modelar : Modelado del desempeño de métodos numéricos en plataformas de hardware heterogéneas	Ernesto Dufrechou y Federico Favaro		https://hdl.handle.net/20.500.12008/53879	Finalizado

Referencias bibliográficas

[1] Top500, lista de Noviembre 2021, https://www.top500.org/lists/top500/2021/11/
[2] Tensor processing unit https://aiyprojects.withgoogle.com/edge-tpu/, Accedido Abr. 2022
[3] M. Ali et al., Level-3 blas on the ti c6678 multi-core DSP, SBAC-PAD, IEEE, 2012.

[4] J. Aliaga et al., Characterizing the efficiency of multicore and manycore processors for the solution of sparse linear systems. Computer Science - R&D, 2016.
[5] J. Aliaga et al., Evaluating the NVIDIA tegra processor as a low-power alternative for sparse GPU computations. En Latin American High Performance Computing
Conference. Springer, Cham, 2017.
[6] M. Amarís et al., A comparison of GPU execution time prediction using machine learning and analytical modeling, NCA, IEEE, 2016.
[7] H. Anzt et al., Preparing sparse solvers for exascale computing, Phil. Trans. R. Soc. A, 2020.
[8] H. Anzt et al., 2020. Load-balancing Sparse Matrix Vector Product Kernels on GPUs, Trans. Parallel Comput. ACM, 2020.
[9] H. Anzt et al., On the performance and energy efficiency of sparse linear algebra on GPUs. Int. J. High Perform. Comput. Appl., 2017.
[10] P. Benner et al., Tuning the Blocksize for Dense Linear Algebra Factorization Routines with the Roofline Model. ICA3PP Workshops, 2016.
[11] P. Benner et al., On the Impact of Optimization on the Time-Power-Energy Balance of Dense Linear Algebra Factorizations. ICA3PP, 2013.
[12] G. Berger, et al., Unleashing the performance of bmSparse for the sparse matrix multiplication in GPUs. ScalA@SC, 2021
[13] B. W. Boehm, Software risk management: principles and practices. IEEE software, 1991.
[14] R. Clint et al. Automatically tuned linear algebra software. SC '98, ACM/IEEE, 1998.
[15] S. Dalton et al., Optimizing sparse matrix operations on gpus using merge path. IPDPS, IEEE, 2015.
[16] L. Díaz, et al., Energy Measurement Laboratory for Heterogeneous Hardware Evaluation, IEEE URUCON, 2021.
[17] P. M. Domingos, A few useful things to know about machine learning. Commun. ACM, ACM, 2012.
[18] E. Dufrechou, P. Ezzatti: Using analysis information in the synchronization-free GPU solution of sparse triangular systems. Concurr. Comput. Pract. Exp. 2020.
[19] E. Dufrechou et al., Machine learning for optimal selection of sparse triangular system solvers on GPUs. J. Parallel Distributed Comput., 2021.
[20] E. Dufrechou et al., Selecting optimal SpMV realizations for GPUs via machine learning. Int. J. High Perform. Comput. Appl., 2021.
[21] E. Dufrechou et al. Automatic Selection of Sparse Triangular Linear System Solvers on GPUs through Machine Learning Techniques. SBAC-PAD, IEEE, 2019.
[22] D. Erguiz et al., Assessing Sparse Triangular Linear System Solvers on GPUs. SBAC-PAD Workshops, IEEE, 2017.
[23] F. Favaro et al., Unleashing the computational power of FPGAs to efficiently perform SPMV operation, SCCC, 2021
[24] F. Favaro et al., Optimizing the performance of sparse linear algebra kernels in FPGAs guided by the Roofline model, bajo evaluación en la revista IEEE Micro
[25] F. Favaro, et al., Hardware Implementation of a Multi-Channel EEG Lossless Compression Algorithm, SPL 2019, Buenos Aires, Argentina, 2019.
[26] F. Favaro et al., Exploring fpga Optimizations to Compute Sparse Numerical Linear Algebra Kernels. ARC2020, 2020.
[27] F. Favaro et al., Understanding the Performance of Elementary Numerical Linear Algebra Kernels in FPGAs, ASHES 2020, New Orleans, USA, 2020.
[28] J. Ferreira el al., A genetic programming approach for construction of surrogate models, in Proceedings of FOCAPD-19, Elsevier, 2019.
[29] G. Golub and C. F. Van Loan., Matrix computations. Vol. 4. JHU Press, 2012.
[30] T. Grutzmacher et al. A customized precision format based on mantissa segmentation for accelerating sparse linear algebra. Concurr. Comput. Pract. Exp., 2019.
[31] J. L. Hennessy and D. A. Patterson, A New Golden Age for Computer Architecture, Communications of the ACM, Feb. 2019.
[32] C. Hong et al. Adaptive sparse tiling for sparse matrix multiplication. PPoPP19, ACM, 2019.
[33] A. Ilic et al. Beyond the Roofline: Cache-Aware Power and Energy-Efficiency Modeling for Multi-Cores, Transactions on Computers, IEEE, 2017.
[34] D. Kirk and W. Hwu, Programming Massively Parallel Processors, Second Edition: A Hands-on Approach, Morgan Kaufmann, 2012.
[35] M. Köhler et al., Energy-aware solution of linear systems with many right hand sides. Computer Science - R&D, 2016.
[36] Y. Liu and B. Schmidt, Lightspmv: Faster cuda-compatible sparse matrix-vector multiplication using compressed sparse rows. Journal of Signal Processing Systems, 2018.
[37] W. Liu et al., Fast synchronization-free algorithms for parallel sparse triangular solves with multiple right-hand sides. Concurr. Comput. Pract. Exp, 2017.
[38] Y. Lu et al., Implementing LU and Cholesky factorizations on artificial intelligence accelerators. CCF Trans. HPC, 2021.
[39] Z. Lu et al., Efficient Block Algorithms for Parallel Sparse Triangular Solve, ICPP, ACM, 2020.
[40] G. Mitra et al., Development and Application of a Hybrid Programming Environment on an ARM/DSP System for High Performance Computing, IPDPS, IEEE, 2018.
[41] R. Marichal, et al., Towards a Lightweight Method to Predict the Performance of Sparse Triangular Solvers on Heterogeneous Hardware Platforms. CARLA, 2019
[42] R. Marichal, et al. Automatic selection of GPU sparse triangular solvers based on energy consumption. PACO-19, Magdeburgo, Alemania, 2019.
[43] F. Montagna, et al., PULP-HD: Accelerating brain-inspired high-dimensional computing on a parallel ultra-low power platform, DAC, IEEE, 2018.
[44] Y. Niu et al., TileSpMV: A Tiled Algorithm for Sparse Matrix-Vector Multiplication on GPUs, IPDPS, IEEE, 2021.
[45] C. Nugteren et al. Roofline-aware DVFS for GPUs, ADAPT '14, ACM, 2014.
[46] J. Silva et al., Balancing Energy and Performance in Dense Linear System Solvers for Hybrid ARM+GPU platforms. CLEI Electron. J., 2016.
[47] V. Volkov, J. Demmel. Benchmarking. GPUs to tune dense linear algebra. SC '08, ACM/IEEE, 2008.
[48] S. Williams, et al., Roofline: an insightful visual performance model for multicore architectures, Commun. ACM, 2009.
[49] M. Winte, et al., Adaptive sparse matrix-matrix multiplication on the GPU. PPoPP, ACM, 2019.
[50] Z. Xie et al., A Pattern-Based SpGEMM Library for Multi-Core and Many-Core Architectures, Transactions on Parallel and Distributed Systems, IEEE, 2022.

[51] M. Freire, E. Dufrechou and P. Ezzatti, "A new level-set analysis and sparse storage format for the SPTRSV in GPUs," 2024 IEEE 36th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD), Hilo, HI, USA, 2024, pp. 59-69, doi: 10.1109/SBAC-PAD63648.2024.00014.
[52] M. Freire, E. Dufrechou and P. Ezzatti, "A synchronization-free incomplete LU factorization for GPUs with level-set analysis," 2025 33rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), Turin, Italy, 2025, pp. 217-225, doi: 10.1109/PDP66500.2025.00037.
[53] Favaro, F.; Dufrechou, E.; Oliver, J.P.; Ezzatti, P. Optimizing the Performance of the Sparse Matrix–Vector Multiplication Kernel in FPGA Guided by the Roofline Model. Micromachines 2023, 14, 2030. https://doi.org/10.3390/mi14112030
[54] Freire, M., Marichal, R., Dufrechou, E., Ezzatti, P. (2023). Towards Reducing Communications in Sparse Matrix Kernels. In: Naiouf, M., Rucci, E., Chichizola, F., De Giusti, L. (eds) Cloud Computing, Big Data & Emerging Topics. JCC-BD&ET 2023. Communications in Computer and Information Science, vol 1828. Springer, Cham. https://doi.org/10.1007/978-3-031-40942-4_2
[55] Mauricio Friss de Kereki, Selección óptima de rutinas de álgebra lineal dispersa usando aprendizaje automático, Proyecto de grado en Ingeniería en Computación, FING, UDELAR
[56] Freire, M. Aplicación de estrategias Synchornization Free a la resolución de operaciones de Algebra Lineal Dispersa. [en línea]. Tesis de maestría. Universidad de la República (Uruguay). Facultad de Ingeniería.. 2025. [Fecha consulta: 4 de marzo 2026].
[57] Berger, G. Uso de formatos de almacenamiento y algoritmos a bloques en álgebra dispersa en dispositivos masivamente paralelos. [en línea]. Tesis de maestría. Universidad de la República (Uruguay). Facultad de Ingeniería.. 2024. [Fecha consulta: 4 de marzo 2026].
[58] Favaro, F. Exploring Field-Programmable Gate Arrays and High-Level Synthesis to perform numerical linear algebra operations for High Performance Computing. [en línea]. Tesis de doctorado. Universidad de la República (Uruguay). Facultad de Ingeniería. 2025. [Fecha consulta: 4 de marzo 2026].