

# Análisis y Modificación de Código Heredado para Cómputo de Alto Rendimiento con Modelos Numéricos del Clima

Fernando G. Tinetti\*, Pedro G. Cajaraville#, Juan C. Labraga###, Mónica A. López###, Diego Rodrigues#

#Departamento de Informática, Facultad de Ingeniería – (UNPSJB<sup>1</sup>)

##Centro Nacional Patagónico (CENPAT-CONICET<sup>2</sup>)

Facultad de Informática – (UNLP<sup>3</sup>)

fernando@info.unlp.edu.ar, gustavo.cajaraville@gmail.com, {labraga, monica}@cenpat.edu.ar, drodrigues2002tw@gmail.com

## RESUMEN

En este proyecto se estudian las alternativas de optimización de un modelo numérico del clima en producción, mediante la utilización de bibliotecas BLAS (*Basic Linear Algebra Subroutines*), las cuales definen un conjunto básico de rutinas que realizan operaciones con vectores y matrices. La optimización es casi inmediata en la mayoría de las implementaciones, con lo cual se obtienen aplicaciones numéricas optimizadas. Aunque el modelo del clima es específico, los modelos numéricos (y sus implementaciones en programas de cómputo), tienden a ser similares en cuanto al tipo de procesamiento (integración o derivación parcial con métodos numéricos) y en cuanto a las dependencias de datos que deben ser tenidas en cuenta para resolver el problema. El producto final de este trabajo aportará al grupo de investigación un modelo climático optimizado para su posterior paralelización. Como mínimo, se tendrá un conocimiento más profundo del modelo y de las operaciones numéricas involucradas. A mediano plazo, el aporte de este proyecto es mucho más *abarcativo* en cuanto a que se tiende a proveer una metodología de optimización de código numérico utilizando BLAS o directamente *recodificando* software en producción. Es importante destacar que el modelo numérico a optimizar ha sido desarrollado con mayor atención en la implementación correcta desde el punto de vista numérico de los procesos físicos modelados que en los demás factores importantes desde el punto de vista de ingeniería de software como lo son las características de rendimiento, y mantenimiento. En este sentido, trabajar sobre software en producción y heredado (o *legacy code*) es todo un desafío en cuanto a que cualquier cambio puede tener efectos colaterales en principio desconocidos.

**Palabras Clave:** optimización de rendimiento, modelos Numéricos, ingeniería inversa, modelado climático, bibliotecas BLAS.

## 1. INTRODUCCIÓN

Este trabajo intenta ser desarrollado dentro del contexto de un proyecto de investigación que consiste en la optimización y paralelización de un modelo climático acoplado océano-atmósfera [1], llevado a cabo por integrantes del CENPAT (Centro Nacional Patagónico)-CONICET, Universidad Nacional de la Patagonia San Juan Bosco y Universidad Nacional de la Plata [4] [9].

El modelo climático a utilizar como ejemplo real de código heredado en producción es el modelo

---

<sup>1</sup> Universidad Nacional de la Patagonia San Juan Bosco

<sup>2</sup> Centro Nacional Patagónico - Consejo Nacional de Investigaciones Científicas y Técnicas

<sup>3</sup> Universidad Nacional de La Plata

\* Investigador Asistente, Comisión de Investigaciones Científicas de la Provincia de Buenos Aires

CSIRO MII, que fue desarrollado en la División de Investigaciones Atmosféricas, CSIRO, Australia [8]. En el año 1999 fue cedido, luego de un período de capacitación, a un grupo de investigación de la Unidad de Investigación de Oceanografía Física y Meteorología (UIOM) del CENPAT-CONICET. La versión cedida incluye un modelo atmosférico, un modelo oceánico y un modelo de la capa de hielo, integrados en un modelo acoplado océano-atmósfera. El código original del modelo CSIRO evolucionó hasta la versión MII, que fue desarrollada para supercomputadoras (CRAY, Silicon Graphics, etc) con varios procesadores bajo sistema operativo UNIX. Para su utilización en la UIOM fue necesario portar el modelo en computadoras de rango intermedio (SUN *workstations* Ultra10, Enterprise) con un solo procesador. Puede decirse que existen en la actualidad tres factores que limitan los avances en el estudio del clima o en la investigación de factores climáticos utilizando el modelo CSIRO MII:

1. Algunos experimentos climáticos que aportarían conocimientos teóricos nuevos sobre el clima, representan un costo computacional muy alto o no son inviables en términos de tiempo de procesamiento.
2. Aplicaciones rutinarias del modelo, como el pronóstico del tiempo o el clima, que requieren numerosas repeticiones de simulaciones de períodos cortos (ej.: varios días a un año) también se ven limitadas en el número de repeticiones posibles cuando forman parte de un pronóstico operativo.
3. Tanto las supercomputadoras como las computadoras de rango intermedio (SUN *workstations*, por ejemplo) tienen un alto costo de mantenimiento y actualización. En general, se ha llegado a que la relación costo/beneficio de las supercomputadoras y las computadoras de rango intermedio y su mantenimiento es muchas veces insostenible en el contexto de las aplicaciones de investigación e inclusive de algunas aplicaciones de producción de bienes y/o servicios. Es por esto que la gran mayoría de las aplicaciones se ha portado a computadoras de escritorio, normalmente PCs con Linux en el entorno científico.

## **1.2 Análisis del Software para la Optimización Secuencial del Modelo.**

Este modelo climático, al igual que muchos otros, se basan en la resolución numérica de un conjunto de ecuaciones diferenciales que expresan las leyes y principios físicos que rigen la dinámica tridimensional de los procesos fundamentales que tienen lugar en cada componente del sistema climático (atmósfera, océano, hielo, nieve, suelo y vegetación), así como los intercambios de energía y masa entre ellos [1]. El complejo sistema no-lineal de ecuaciones diferenciales resultante, se resuelve dividiendo el espacio ocupado por la atmósfera y el océano en pequeñas celdas tridimensionales. En cada una de ellas se asignan valores a variables que caracterizan su estado (temperatura, densidad, velocidad, etc.) a partir de observaciones directas o indirectas en un determinado instante inicial. Comenzando con estos valores, se resuelven las ecuaciones para derivar las evoluciones temporales de dichas variables de estado en cada celda de la matriz del modelo. Esto se hace calculando iterativamente los valores previstos de tales variables en intervalos temporales discretos (paso temporal), es decir avanzando en el tiempo hasta llegar al final del período de simulación que se desee.

Procesar el inmenso volumen de datos involucrado por los modelos numéricos en general, requiere realizar miles de millones de operaciones aritméticas en cada intervalo temporal, hasta completar el período de simulación, que en algunas ocasiones abarca centenares de años. Estas simulaciones se deben completar en un período de tiempo razonable, lo que requiere una eficiente optimización secuencial del modelo. Una de las formas de llevar a cabo esta optimización (al menos en términos teóricos), reemplazando las rutinas de multiplicación vector-matriz y/o matriz-matriz del modelo original, por funciones de biblioteca que respondan a las especificaciones de BLAS (Basic Linear Algebra Subroutines) [2] [3]. La biblioteca BLAS define un conjunto básico de rutinas, para realizar

operaciones con vectores y matrices que se puede implementar eficientemente. Existen distintas implementaciones de BLAS en producción, que eventualmente se podrían aprovechar en el contexto del modelo climático. Estas implementaciones incluyen ACML (AMD Core Math Library) [5], MKL (Math Kernel Library de Intel) [11], y ATLAS (Automatically Tuned Linear Algebra Software) [6].

Si bien se podrían utilizar o desarrollar herramientas de ingeniería inversa, el objetivo inmediato de este proyecto es el de optimizar rendimiento, por lo tanto se considera que las herramientas y metodologías de ingeniería inversa son demasiado generales. En principio, tampoco hay disponible información ni herramientas específicas para optimizar código heredado del que no se tiene toda la documentación disponible. De hecho, se tiene el modelo CSIRO MII en código fuente Fortran 77 y algunos reportes técnicos de los procesos físicos modelados. Esto significa que de alguna manera se debe analizar exhaustivamente el código fuente disponible con el objetivo de *adaptarlo* para utilizar la biblioteca BLAS o directamente optimizar los patrones de procesamiento para el hardware disponible.

## 2. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

En principio, se podrían enumerar algunas de las tareas involucradas en el desarrollo del proyecto de optimización:

1. Dado que la optimización de rendimiento está enmarcada en un proyecto de mayor envergadura, justamente el primer paso es el estudio de este proyecto. Se debe remarcar que no se estudiará desde la perspectiva del área de procesos climáticos sino desde la perspectiva de procesamiento numérico y de la optimización de rendimiento.
2. Estudiar la especificación de BLAS y las diferencias y/o similitudes de las distintas implementaciones disponibles, como ACML, MKL y ATLAS.
3. Estudiar las alternativas para la construcción de un entorno de experimentación y desarrollo de software numérico, en principio orientado al modelo climatológico. Esto involucra el análisis de alternativas de:
  - 3.1. Software de base como los operativos de 32 o 64 bits, que pueden tener diferencias importantes en el área de procesamiento numérico.
  - 3.2. Compiladores a utilizar, como el compilador de Fortran de Intel *ifort* u otros compiladores de Fortran 77 o Fortran 90/95 como *g77* o *gfortran*.
  - 3.3. Implementaciones disponibles de BLAS, como ACML, MKL y ATLAS.
4. Analizar el rendimiento de las diferentes implementaciones de BLAS, independientemente del modelo numérico, que proporciona una forma de ganar experiencia y de comparar las posibilidades de optimización.
5. Detectar los módulos del modelo climático son los que consumen mayor tiempo de procesamiento. Esto implica utilizar herramientas al menos de *profiling* de ejecución y también proporciona experiencia en la utilización del software (básicamente en términos de parámetros y datos de entrada y de salida).
6. Investigar sobre el lenguaje Fortran [10] y sus diferentes variantes: Fortran 77, Fortran 90/95, con el objetivo de entender el código existente en Fortran 77 y analizar la posibilidad de modificarlo o actualizarlo a Fortran 90/95.
7. Modificar el código original de los módulos que mayor tiempo de procesamiento requieren, reemplazándolo por funciones BLAS en todos los puntos donde sea posible.
8. Comparar los resultados obtenidos con la ejecución del modelo original con los obtenidos con la ejecución del modelo modificado.

9. Analizar las posibilidades de aplicación de la metodología a otros modelos numéricos de procesos físicos con las mismas características.

### 3. RESULTADOS OBTENIDOS/ESPERADOS

Ya se cuenta con un análisis preliminar de rendimiento de implementaciones de BLAS y, en cierto sentido, información *relevante* de las posibilidades de optimización de código fuente Fortran 77. La Tabla 1 muestra el rendimiento en Mflop/s (millones de operaciones de punto flotante por segundo) de una de las rutinas más representativas de BLAS: DGEMM (multiplicación de matrices) con distintas alternativas en la misma computadora para matrices cuadradas de orden 3000. A partir de los resultados de la Tabla 1 se tiene una primera idea de que la optimización no solamente es posible sino que aportaría una ganancia muy importante de rendimiento, ya que la rutina sin optimizar tiene menos de 20 veces el rendimiento de todas las demás alternativas (usando diferentes implementaciones disponibles). El compilador utilizado es *ifort* (de Intel), el sistema operativo es Linux de 32 bits y el hardware de procesamiento es una PC de escritorio con microprocesador AMD Athlon +3000. La Fig. 1 muestra los mismos datos en forma gráfica.

Implementación	Mflop/s
Sin optimizar	104
ATLAS	2.429
ACML	2.773
MKL	2.524

Tabla 1: Rendimiento de DGEMM con Diferentes Bibliotecas.

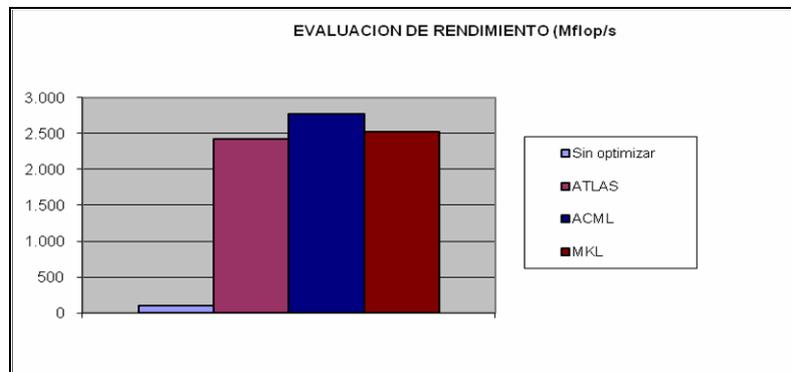


Figura 1: Comparación de rendimiento de DGEMM.

Básicamente, para poder ampliar el campo de aplicación de este modelo climático (y probablemente de otros modelos existentes), el proyecto de investigación general [WICC07] intentará alcanzar dos objetivos principales:

- 1º) Optimizar el modelo en forma secuencial.
- 2º) Optimizar el modelo en forma paralela.

El primer objetivo, intentará cumplirse con el desarrollo de este proyecto, el cual consistirá en estudiar la posibilidad de optimización del modelo secuencial existente mediante el uso de bibliotecas BLAS (Basic Linear Algebra Subroutines) [7] [2] [3]. Las implementaciones de esta biblioteca o bibliotecas similares, usualmente permiten el incremento significativo del rendimiento de los microprocesadores, debido a que utilizan eficientemente los recursos de cómputo disponible, tales como la memoria *cache* de los mismos [6] [12]. La consecución de este objetivo aportará al grupo de investigación un modelo climático optimizado para su posterior paralelización. Como

mínimo, se tendrá un conocimiento más profundo del modelo y de las operaciones numéricas involucradas.

Como otros resultados relacionados con el de la optimización de modelo numérico (aunque de menor importancia) se podrían enumerar:

- Documentación de numerosas opciones de implementación de bibliotecas y la inserción de las mismas dentro de modelos numéricos ya desarrollados y en producción.
- Identificación de patrones de procesamiento que existen en el modelo climático pero que se pueden aplicar a otros modelos numéricos.
- Documentación y comparación de las opciones de procesamiento numérico de bajo costo (sobre PCs) en cuanto a sistemas operativos (de 32 o 64 bits), compiladores y bibliotecas de cómputo numérico disponibles.

## 4. BIBLIOGRAFÍA

[1] M. de Castro, “El Modelado del Clima Terrestre”, [http://grupos.unican.es/ai/meteo/cursos/2007\\_cursoRegionalizacion\\_files/2\\_1\\_Castro\\_resumen.pdf](http://grupos.unican.es/ai/meteo/cursos/2007_cursoRegionalizacion_files/2_1_Castro_resumen.pdf)

[2] Dongarra J., J. Du Croz, S. Hammarling, R. Hanson, “An extended Set of Fortran Basic Linear Subroutines”, ACM Trans. Math. Soft., 14 (1), pp. 1-17, 1988.

[3] Lawson C., R. Hanson, D. Kincaid, F. Krogh, “Basic Linear Algebra Subprograms for Fortran Usage”, ACM Transactions on Mathematical Software 5, pp. 308-323, 1979.

[4] F. G. Tinetti, P. G. Cajaraville, J. C. Labraga, M. A. López, "Cómputo Paralelo Aplicado a Modelos Numéricos del Clima", IX Workshop de Investigadores en Ciencias de la Computación, Universidad Nacional de La Patagonia San Juan Bosco (UNPSJB), Trelew, Chubut, Argentina, Mayo 3-4 de 2007.

[5] AMD Core Math Library (ACML), <http://developer.amd.com/acml.jsp>

[6] ATLAS (Automatically Tuned Linear Algebra Software), [www.netlib.org/atlas/index.html](http://www.netlib.org/atlas/index.html)

[7] BLAS (Basic Linear Algebra Subprograms), <http://www.netlib.org/blas/index.html>

[8] Commonwealth Scientific and Industrial Research Organisation (CSIRO), <http://www.csiro.au/>

[9] Cómputo de Alto Rendimiento - Aplicaciones de Algebra Lineal, <http://hpclinalg.googlepages.com/>

[10] Intel Fortran Compiler Documentation, <http://www.ncsa.uiuc.edu/UserInfo/Resources/Software/Intel/Compilers/8.1/fcompindex.htm>

[11] Intel Math Kernel Library, <http://www.intel.com/cd/software/products/asmo-na/eng/perflib/307757.htm>

[12] PHiPAC Fast Matrix Multiply Home Page, <http://www.icsi.berkeley.edu/~bilmes/hipac/>