

Aplicaciones Científicas Numéricas: El (Ciclo de Vida del) Software Heredado

Mariano Méndez, Fernando G. Tinetti¹

Instituto de Investigación en Informática LIDI (III-LIDI)
Facultad de Informática – UNLP

fernando@lidi.info.unlp.edu.ar, marianomendez@gmail.com

RESUMEN

El objetivo de esta línea de investigación es el estudio de las aplicaciones científicas de cómputo numérico desde el punto de vista de la Ingeniería de Software y del Procesamiento Paralelo. Además, se orienta a desarrollar técnicas y herramientas destinadas a reestructurar dichas aplicaciones permitiendo que éstas evolucionen a través del tiempo mejorando sus prestaciones y su estructura.

CONTEXTO

Esta línea de Investigación forma parte de las tareas de investigación del proyecto "Proyecto de Incentivos 11/F010 Arquitecturas Multiprocesador Distribuidas: Modelos, Software de Base y Aplicaciones" del Instituto de Investigación en Informática LIDI acreditado por la UNLP. El planteo que se presenta constituye una línea incipiente en el III-LIDI

Palabras Claves: Software heredado, Reestructuración de Código, Aplicaciones Científicas, Paralelización.

1. INTRODUCCION

Las aplicaciones científicas numéricas conforman el tipo de software más antiguo que ha sido desarrollado, sus orígenes datan al menos desde 1950 [1]. Gracias a los avances en los procesos de desarrollo de software, podría pensarse que este tipo de aplicaciones ha evolucionado paralelamente con dichas técnicas de desarrollo de software. Esta suposición dista mucho de la realidad en la que se encuentra este tipo de software. Existe un gran número de áreas de la ciencia que utiliza estas aplicaciones como herramientas para sus investigaciones, algunas de las más relevantes en la actualidad son: Climatología, Cristalografía, Física de partículas elementales, Química, Ingeniería, Geología, Astronomía, etc.

Un gran ejemplo de este tipo de aplicaciones es el de los llamados Modelos Climáticos Globales, utilizados para realizar predicciones, generalmente relacionadas con el calentamiento global del planeta.

Muchas de estas aplicaciones se caracterizan por:

¹ Investigador CIC Pcia. De Buenos Aires.

- Haber sido construidas sin la utilización de un proceso de desarrollo de software.
- Su operación es muy compleja. No existe documentación o es inadecuada.
- Han sido desarrolladas con técnicas de programación ampliamente superadas. En general, existe un gran porcentaje que desaprovecha el cómputo paralelo en en multicores.
- Son aplicaciones que dependen fuertemente de sus creadores.

El estudio de este tipo de aplicaciones es necesario para determinar cuáles son los aportes que la Ingeniería de Software puede proporcionar a cada una de las disciplinas que utilizan y producen este tipo de aplicaciones. Además, cabe destacar que existen disciplinas en las cuales este tipo de software es primordial debido a la dramática importancia que éstas han adquirido, por ejemplo en la meteorología y el estudio del impacto del cambio climático.

Esta clase de aplicaciones posee un conjunto muy complejo de requerimientos funcionales, como así también un conjunto de requerimientos no funcionales (rendimiento, por ejemplo) que deben ser estrictamente cumplidos para que la misma arroje resultados útiles. Bajo este punto de vista, es fundamental poder agregarles nuevas técnicas de programación que permitan el cumplimiento de dichos requerimientos no funcionales. Aquí es donde la programación en paralelo ya sea orientada a paralelismo con memoria compartida o paralelismo en computadoras paralelas con memoria distribuida adquiere vital importancia.

La introducción de todas estas técnicas dentro de la estructura interna de este tipo de aplicaciones es muy compleja. Para

ello es necesario tener conocimientos de varias disciplinas relacionadas con las Ciencias de la Computación. Algunas de estas áreas son: Ing. de Software, Programación para Cómputo Paralelo, Ing. de Requerimientos, etc.

Hacia 1990 se calculaba que la cantidad de millones de líneas de código de software heredado llegaba a 120.000 millones de líneas de código. [2]. En el año 2000 se calcula que había unos 250.000 millones de líneas de código escritas [3]. Existen varias aproximaciones del costo que representa dentro del ciclo de vida de una aplicación la etapa de mantenimiento, el mismo varía entre el 70% [4] y el 90 % [5] del total de los recursos disponibles.

2. LINEAS DE INVESTIGACION Y DESARROLLO

Se plantean como temas de estudio:

- Proceso de Mantenimiento de software heredado: en la actualidad existe poca formalización de los procesos de desarrollo destinados a la introducción de cambios y mejoras en las etapas de mantenimiento de software. Teniendo en cuenta que este proceso es uno de los que más recursos insumen dentro del ciclo de vida de una aplicación (hasta el 70 % de los recursos totales [5]), es de vital importancia la formalización de procesos de desarrollo para esta etapa del ciclo de vida de un proyecto. Se está investigando en dirección a la incorporación de metodologías ágiles en las tareas de mantenimiento. Una de las alternativas analizadas en la actualidad se muestra esquemáticamente en la Fig. 1.

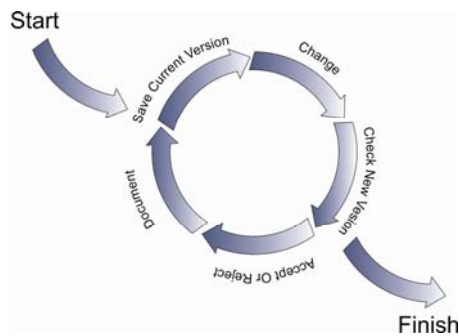


Figura 1: Proceso Incremental de Cambios para Software Heredado.

- **Paralelización:** uno de los objetivos más importantes de esta línea de investigación es la implementación de una herramienta capaz de paralelizar programas secuenciales en forma automática o al menos asistida. Se están estudiando un conjunto de transformaciones de código fuente destinadas a convertir secciones de código secuencial en paralelo. Para ello se estudian el conjunto de Pre y Post condiciones (que podrían llegar a ser independientes del lenguaje de programación) que el mismo debe cumplir. En la actualidad, dicha herramienta se encuentra en fases tempranas de implementación, se está desarrollando una prueba de concepto utilizando el entorno de programación Eclipse.
- **Estudio específico de la estructura interna del software científico heredado:** se ha seleccionado un conjunto de aplicaciones científicas pertenecientes a distintos Modelos Climáticos Globales. Se estudian las características más relevantes de este tipo de software científico, con el objetivo de determinar los posibles puntos de mejoras en la estructura interna de los mismos. La modernización de estos programas ayudarán a aumentar la precisión de las predicciones realizadas mediante el análisis de los resultados por parte de los científicos [6].
- **Métricas:** Se está realizando un estudio cuantitativo y comparativo de la estructura interna de las aplicaciones científicas heredadas, más específicamente del software científico escrito en Fortran. Se intentan determinar estilos de programación, estándares más utilizados, etc. Para ello se está estudiando qué métricas aplicar en más de 500 MB de código fuente escrito en Fortran y disponible en internet.
- **Estudio comparativo de herramientas:** la función que desempeñan las herramientas que automatizan las tareas de mantenimiento de software ha cobrado vital importancia en los últimos tiempos. Se están realizando estudios comparativos de herramientas destinadas a las tareas de mantenimiento.
- **Implementación de nuevas herramientas:** Se están estudiando distintos *frameworks* para el desarrollo de herramientas que permitan aplicar transformaciones al código fuente Fortran mediante la manipulación de AST (Abstract Syntax Tree), como por ejemplo ROSE y Photran.
- **Casos de estudio:** Actualmente se están desarrollando tareas de reestructuración y paralelización en distintas aplicaciones en colaboración con otras instituciones a nivel internacional.

3. RESULTADOS OBTENIDOS/ESPERADOS

Los resultados (tanto los obtenidos como los esperados a mediano y largo plazo) tienen varias orientaciones:

- De acuerdo con las tareas desarrolladas y a desarrollar, los resultados se

enfocarán en varias direcciones relacionadas con el software heredado y un poco más específicamente con la paralelización del mismo.

- Se han documentado un conjunto de reestructuraciones de código Fortran específicamente. Se ha detallado un catálogo y una posible clasificación [7].
- Ya se cuenta con un conjunto de reestructuraciones implementadas, detalladas y documentadas en [7]. Algunas de las mismas ya están incluidas en el Photran, un plug-in de Eclipse [8] específicamente orientado a Fortran. La Fig. 2 muestra una de las posibles transformaciones ya implementadas.

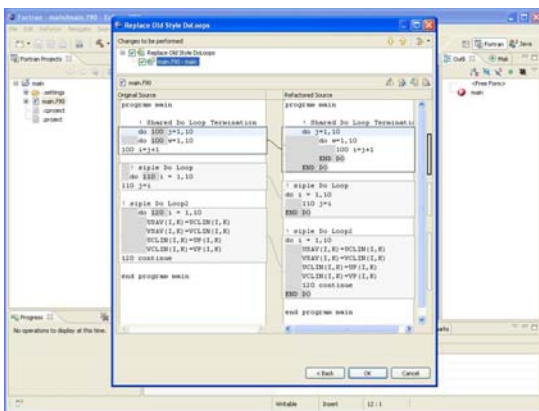


Figura 2: Una de las Transformaciones de Código Implementadas en Photran.

- Se está trabajando en la implementación de una prueba de concepto orientada a paralelismo con memoria compartida que utilice el estándar de OpenMp.
- Es está trabajando en la especificación de un catálogo de transformaciones específicas destinadas a la paralelización de código fuente. Determinación de las pre y post condiciones de di-

chas transformaciones. Se está estudiando la posibilidad de tener esta especificación independiente del lenguaje de programación.

- Se está trabajando en la construcción, verificación y aplicación de un modelo de proceso de mantenimiento específico para aplicaciones heredadas. Más específicamente, en el proceso mostrado en la Fig. 1.
- Se está estudiando las formas en que se pueden determinar y describir las tareas y actividades básicas necesarias para la aplicación del proceso de mantenimiento de software heredado.
- Se busca determinar una medición de las mejoras aportadas por el proceso y las herramientas desde el punto de vista de la ingeniería de software y de la ganancia en rendimiento de las aplicaciones. Específicamente, se está trabajando en la determinación de una métrica para las mejoras obtenidas por transformación del código fuente, tal como la propuesta en [7].
- Se busca aportar mejoras concretas a software científico actualmente en producción mediante la aplicación de las transformaciones y procesos mencionados anteriormente.
- También se está trabajando en la propuesta de un “tool-chain” de código abierto específico para la aplicación del proceso de reestructuración para el software heredado. Este tool-chain estará compuesto por un conjunto de herramientas que puedan satisfacer cada etapa del proceso.

4. FORMACION DE RECURSOS HUMANOS

Ya se ha completado una Especialización y una Maestría en Ingeniería de Software

directamente relacionadas con los temas mencionados. Actualmente se está en la etapa de definición de al menos una tesis doctoral y una tesina de licenciatura, en trabajos con alumnos de grado avanzados.

5. BIBLIOGRAFIA

1. Backus, J. "The history of Fortran I, II, and III". 1978. ACM. History of programming languages I. pp. 25-74,
2. Ulrich, W. "The evolutionary growth of software engineering and the decade ahead". 1990. American Programmer 3 (10), 12-20.
3. Sommerville, I. "Software Engineering (6th Edition)". 2000. Addison-Wesley.
4. Eastwood, A. "Firm fires shots at legacy systems". 1993. Computing Canada 19 (2), p. 17.1
5. Erlikh, L. "Leveraging legacy system dollars for E-business". 2000. (IEEE) IT Pro, May/June, pp. 17-23.
6. Taylor, K.E. and Stouffer, R.J. and Meehl, G.A." A summary of the CMIP5 experiment design ". World. December 2009 . p.1-33
7. Méndez M. "Fortran Refactoring for Legacy Systems". 2011. Master Thesis. UNLP.
8. Photran - An Integrated Development Environment and Refactoring Tool for Fortran, <http://www.eclipse.org/photran/>