

## **Resumen de la tesis**

### **Sintonización Dinámica de Aplicaciones Paralelas/Distribuidas**

Anna Morajko

El objetivo principal de las aplicaciones paralelas/distribuidas es resolver el problema considerado del modo más rápido posible utilizando los recursos disponibles. Por lo tanto, el rendimiento se convierte en uno de los aspectos más importantes. El método clásico de mejorar el rendimiento se basa en el análisis de la información de rendimiento obtenida durante la ejecución de la aplicación. Los programadores deben analizar dicha información buscando problemas de rendimiento, determinando las causas de dichos problemas y cambiando el código fuente manualmente. Esta aproximación requiere un elevado grado de experiencia en programación paralela para ser llevada a cabo de modo eficiente. Por lo tanto, el método clásico de sintonización de aplicaciones es una tarea difícil especialmente para programadores no expertos. Además, esta aproximación no es fiable cuando las aplicaciones o los entornos de ejecución tienen un comportamiento dinámico. Muchas aplicaciones tienen un comportamiento diferente según los datos de entrada o incluso pueden variar durante la misma ejecución. Así pues, es necesario desarrollar nuevas aproximaciones que mejoren el rendimiento sin la intervención del usuario y adapten el comportamiento de la aplicación a las condiciones dinámicas que se puedan presentar en tiempo de ejecución.

Esta tesis aborda el problema de la sintonización automática y dinámica de aplicaciones paralelas y distribuidas. Nuestro objetivo es ayudar a los programadores en todo el proceso de mejorar rendimiento de las aplicaciones. La solución propuesta incluye una solución que trata con asuntos de una mejora de aplicaciones dinámica y automática. En esta aproximación, una aplicación es monitorizada, se detectan sus problemas de rendimiento, se buscan las soluciones y finalmente se la sintoniza en tiempo de ejecución. Todos estos pasos son realizados de forma automática, dinámica y continua durante la ejecución. Por tanto, los programadores únicamente deben centrarse en el desarrollo de la aplicación y quedan liberados de las fases relacionadas con el análisis de rendimiento.

Con este objetivo se ha desarrollado un entorno denominado MATE (Monitoring, Analysis and Tuning Environment) que proporciona tales funcionalidades. MATE soporta una sintonización dinámica haciendo tres fases básicas y continuas: monitorización, análisis de rendimiento y modificaciones. Este entorno instrumenta una aplicación durante el tiempo de ejecución de forma dinámica y automática para obtener información sobre el comportamiento de dicha aplicación. La fase de análisis busca los problemas, detecta sus causas y proporciona las soluciones para eliminar esos problemas de rendimiento. Finalmente, MATE sintoniza la aplicación aplicando las soluciones dinámicamente. Además, mientras la aplicación está siendo sintonizada, no necesita ser compilada ni ejecutada otra vez.

El entorno ha sido probado con un amplio conjunto de aplicaciones paralelas y distribuidas para ver si esta aproximación realmente funciona. Hemos comprobado que la metodología de sintonización dinámica es eficaz, fiable, beneficiosa y que puede ser usada para mejoras reales del rendimiento de las aplicaciones. La ejecución de una aplicación bajo control del sistema dinámico ha permitido la adaptación de su comportamiento a las condiciones actuales y la mejora de su funcionamiento.

## **Thesis resume**

### **Dynamic Tuning of Parallel/Distributed Applications**

Anna Morajko

The main goal of parallel/distributed applications is to solve the considered problem as fast as possible utilizing a certain minimum of the parallel system capacities. In this context, the application performance is one of the most important issues. The classical way of improving the application performance is based on the analysis of the monitoring information obtained from an execution of the application. Developers must search through this information for the bottlenecks and optimize the application behavior changing the source code manually. This approach requires developers to do many tasks and have a great experience about parallel programming. Therefore, the classical application tuning is then very difficult especially for non-expert programmers. It is necessary to provide tools that automatically carry out these tasks. Moreover, this classical approach is not feasible when the applications have a dynamic behavior. Many applications have a different behavior according to the input data set or even change their behavior dynamically during the execution. In this case, another approach is required to accomplish performance expectations. It would be desirable that the performance tuning could be done on the fly by modifying the application according to the particular conditions of the execution.

This thesis addresses the problem of automatic and dynamic tuning of parallel and distributed applications. Our objective is to help developers in the process of improving the application performance. This work presents a whole solution that deals with the issues of automatic and dynamic application improvement. In this approach, an application is monitored, its performance bottlenecks are detected, solutions are given and the application is modified on the fly. All these steps are performed automatically, dynamically and continuously during application execution. This approach exempts developers from performance analysis and difficult intervention to a source code by automatically improving the performance of parallel programs during run-time. The dynamic analysis and introduced modifications permits to adapt the behavior of the application to dynamic variations.

With this objective we have developed an environment called MATE (Monitoring, Analysis and Tuning Environment) that provides dynamic automatic tuning of parallel applications. MATE performs dynamic tuning in three basic and continuous phases: monitoring, performance analysis and modifications. This environment dynamically and automatically instruments and traces a running application to gather information about the application behavior. The analysis phase searches for bottlenecks, detects their causes and gives solutions on how to overcome them. Finally, the application is dynamically tuned by applying given solution. Moreover, while it is being tuned, the application does not need to be re-compiled, re-linked and restarted.

Many various practical experiments have been conducted on distributed and parallel applications to see if this approach really works. We have proven that it is effective, feasible, profitable, and can be used for a real improvement of the program performance. Running applications under control of a dynamic tuning system has allowed for adapting their behavior to the existing conditions and improving their functionality.