

Opportunities for Global Optimization: Breaking the Boundaries Across System Components

Ramon Bertran, Marisa Gil, Javier Cabezas, Victor Jimenez,
Lluís Vilanova, Enric Morancho, Nacho Navarro
Computer Architecture Department, Technical University of Catalonia
{rbertran,marisa,jcabezas,victorjj,vilanova,enricm,nacho}@ac.upc.edu

March 12, 2006

Abstract

Usually, optimization techniques are applied separately to each component of the computer system (such as *applications, libraries and operating system*) without taking into account the interaction between them. This approach is useful for general-purpose systems that have to maintain compatibility with several hardware and software components. However, the specialized systems, typically built from a general purpose one, have a reduced and fixed set of components (for instance, some embedded systems), therefore this approach does not produce a system that fits well the requirements for this kind of devices (specially memory footprint and size on persistent storage medium, but sometimes execution time too). Hence, current system developers have to manually tune the system and remove unused functionalities.

In this work we present some optimization opportunities that arise when all the components in the system are taken into account in a single global view, performing optimizations on all of them at the same time. As an example, we show how constant propagation and dead code elimination could be extensively and globally applied on such systems. For this purpose, a previous study to know how the components of the system interact has been done, identifying entry points and inter-component calls for each component in the system. This information will let us to build a global control flowgraph of all the software components to optimize.

We have studied two systems with different characteristics: Linux and L4. Comparing these two completely different kernel paradigms we can define under which conditions certain optimizations can be applied.

- Linux is a monolithic kernel which runs entirely in privileged mode. The interconnections between the system components are basically system calls. Therefore, targeting an specific embedded system, the opportunities appear by removing unused (globally disconnected) system calls, and optimizing kernel code by propagating constant system call parameters from the applications to the kernel.
- Our experience with the L4 microkernel shows that this low level thin components has been tuned by their implementers to a level that to apply new optimizations to them is very hard. As communications between clients and servers rely on IPC messages which have to go through the kernel, we have focused on the opportunities that arise when properties (invariants, alias, etc) are propagated across the communication path.

In conclusion, we have introduced a new approach to optimize the full system by tailoring it for its requirements. This way, we find out more optimization cases, so we finally get a more specialized system than customizing separately. As results, we will show the opportunities on a web server (only 98 of 277 system calls are used on Linux) and a file server systems. We also conclude that the system design (how the components are related) affects the optimizations that could be applied afterwards. Future trends will be to enhance tools to generate automatically the global flowgraph construction and specialize the optimizations for each particular system architecture.

This work has been supported by the Ministry of Science and Technology of Spain and by the European Union (FEDER) under contract TIC2001-0995-C02-01 and the HiPEAC European Network of Excellence. All products or company names mentioned herein are the trademarks or registered trademarks of their respective owners.