



# Challenges in Binary Translation for Desktop Supercomputing

Rodrigo Dominguez, David Kaeli

► **To cite this version:**

Rodrigo Dominguez, David Kaeli. Challenges in Binary Translation for Desktop Supercomputing. AMAS-BT - 3rd Workshop on Architectural and Microarchitectural Support for Binary Translation, Jun 2010, Saint Malo, France. <inria-00492920>

**HAL Id: inria-00492920**

**<https://hal.inria.fr/inria-00492920>**

Submitted on 17 Jun 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Challenges in Binary Translation for Desktop Supercomputing

Rodrigo Dominguez  
David Kaeli

Computer Architecture Research Laboratory  
Northeastern University  
Boston, MA

## Invited Paper

### Abstract:

Given the fact that the microprocessor industry has jumped off the frequency scaling bandwagon due to power issues, the rate of development of aggressive many-core systems has picked up in the past few years. However, these many-core systems have taken a range of incarnations: graphics processors (GPUs) have emerged as the data-parallel architecture of choice; heterogeneous architectures (e.g., AMD Fusion and IBM Cell) have emerged, though present challenges to code developers and compiler writers; and general purpose CPUs continue to increase the number of cores based on Moore's Law, though they lack the data-parallel horsepower found on other architectural styles.

These many-core systems come equipped with a range of programming languages, runtime environments, and compiler frameworks. In the case of GPUs, the architecture of the programmable shader core continues to evolve today in an effort to adapt these graphics-oriented processors to the requirements of the scientific computing community. For this reason, both AMD and NVIDIA have defined intermediate representations (IRs) as part of their offerings. The main role of any IR is to provide a stable instruction set architecture that spans multiple microarchitecture generations, similar to the concept of bytecodes in Java. In the case of AMD/ATI GPUs, the IR is called the Intermediate Language (IL), and for NVIDIA GPUs, the IR is called Parallel Thread Execution (PTX).

Our work explores the fundamental differences between PTX and IL and the benefits of each IR. We are enhancing a binary translation framework to translate PTX into IL, allowing applications compiled for NVIDIA's C for CUDA environment targeting NVIDIA GPUs to run on AMD/ATI GPUs. We will report on our results to date, and will shed some light on the challenges that lay before us.