

How to attack the multi-many-core programmer productivity wall?

The joint research center

BSC and Microsoft extended their two year collaboration and created a joint research center which opened on 18 January 2008 in Barcelona.

The main challenge in designing future processors is how to make them easy to program. The research center will tackle this challenge and aim to have an impact on the way processors for the mobile and desktop market segments are designed ten years from now and beyond.

Areas of expertise:

- Computer Architecture
- Programming Languages

Stakeholders:

- Barcelona Supercomputing Center
- Microsoft Research Cambridge

Two main areas of research:

- Hardware and Software Transactional Memory (TM)
- Hardware support for runtimes

TM programming models and implementation

Extensions to facilitate TM models:

- C Atomic extension
- Extending OpenMP with TM
- Haskell

Research on:

- Software Transactional Memory (STM) implementations
- Full Hardware Transactional Memory (HTM)
- Hardware accelerated STM

Benchmarks/Applications:

- Functional: BSC Haskell STM benchmark suite
- Imperative: C and C++

Infrastructure development:

- Source-to-source restructuring tools
- Runtime libraries
- Architecture simulators

TM application profiling environment

To analyze the results, the Haskell STM runtime was instrumented with hardware performance counters. This framework is general enough and is applied to other in-house TM systems.

This analysis framework can provide results on all aspects of transactional behavior such as commit time, abort overheads and read/write set sizes.

Hardware support for runtimes

Managed languages such as C# or Haskell offer flexibility in return for performance overhead, we are looking at increasing performance through hardware support.

Runtime features such as garbage collection, dynamic recompilation or speculative parallelization are candidates for performance optimization.

Requires tight collaboration between computer architecture and language/runtime.

Reliability and application debugging are other research targets in this area.

