EIGHTH ANNUAL IEEE/ACM INTERNATIONAL SYMPOSIUM ON CODE GENERATION AND OPTIMIZATION (CGO 2010)

April 24-28, 2010
Toronto, Ontario, Canada

CALL FOR PAPERS

Sponsored by IEEE Computer Society tc-μArch and ACM SIGMICRO

The International Symposium on Code Generation and Optimization (CGO) provides a premier venue to bring together researchers and practitioners working at the interface of hardware and software on a wide range of optimization and code generation techniques and related issues. The conference spans the spectrum from purely static to fully dynamic approaches, including techniques ranging from pure software-based methods to architectural features and support. Original research contributions are solicited in areas including but not limited to the following:

- Compilers, back-end code generators, translators, binary optimization tools and runtime environments; static, dynamic, adaptive, or continuous techniques
- New or improved optimization algorithms, including profile-guided and feedback-directed optimization
- Thread extraction and thread-level speculation, especially for multi-core and many-core systems
- Analyses, and optimizations targeting heterogeneous processors and/or GPUs
- Virtualization support for multicore and/or heterogeneous computing
- Phase detection and analysis techniques
- Language features and runtime support for parallelism (including support for transactional semantics, efficient message passing, and dynamic thread creation)
- Program characterization methods targeted at program optimization
- Code transformations to address security, reliability, virtualization, temperature, or energy efficiency
- Architectural support for improved profiling, optimization and code generation
- Experiences with real dynamic optimization and compilation systems on general purpose, embedded system and HPC platforms
- Library and system call support for optimization and code generation
- Solutions that involve cross-layer (HW/OS/VM/SW) design integration
- Efficient profiling and instrumentation techniques
- Memory management, including data distribution, synchronization and garbage collection
- Intermediate representations that enable more powerful or efficient optimization
- Traditional compiler optimizations

IMPORTANT DATES:
Abstracts: September 3, 2009
Papers: September 10, 2009, 11:59PM EDT
Response: November 11, 2009

http://www.cgo.org/