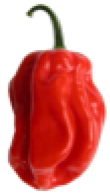
Research Projects

Habanero team members are currently engaged in the following projects to realize the vision of creating portable parallel abstractions for future multicore hardware with high productivity and high performance. Please see the Habanero Publications page for more details:



- Parallel programming models: Habanero Java, Habanero C, Habanero Scala, Habanero Concurrent Collections, Habanero-C++, Habanero-UPC++
- NSF Expeditions Center for Domain-Specific Computing (CDSC)
- Dynamic Exascale Global Address Space Programming Environments (DEGAS)
- Traleika Glacier X-Stack Program
- DSL Technology for Exascale Computing (D-TEC)
- Open Community Runtime (OCR)

Broadly, the research scope for the Habanero team includes the following topics that build on our past work:

1) Language research

1. • Explicit parallelism: portable constructs for homogeneous & heterogeneous multicore
2. • Implicit deterministic parallelism: array views, single-assignment constructs
3. • Implicit non-deterministic parallelism: unordered iterators, partially ordered statement blocks
4. • Builds on our experiences with the X10, CAF, HPF, Fortran 90, and Sisal languages

2) Compiler research

1. New Parallel Intermediate Representation (PIR)
2. Analysis and transformation framework for the PIR
3. Optimization of high-level arrays and iterators
4. Optimization of synchronization, data transfer, and transactional memory operations
5. Code partitioning for accelerators

Builds on our experiences with the D System, Massively Scalar, Telescoping Languages Framework, ASTI, and PTRAN research compilers

3) Virtual machine research

1. VM support for work-stealing scheduling algorithms with extensions for places, transactions, task groups
2. Integration and exploitation of lightweight profiling in VM scheduler and memory management system
3. Builds on our experiences with the Jikes Research Virtual Machine

4) Parallel runtime system research

1. Extensions to work-stealing task scheduling algorithms
2. Fine-grained producer-consumer synchronization (phasers)
3. New nonblocking data structures
4. Efficient software transactional memory libraries
5. Builds on our experiences with the java.util.concurrent and the DSTM2 libraries

5) Tools research

1. Program analysis for common parallel software errors
2. Performance attribution of shared code regions (loops, procedure calls) using static and dynamic calling context
3. Builds on our experiences with the Rice HPCToolkit, Eclipse Parallel Tools Platform, and DrJava projects