OPENSHMEM ANALYZER (OSA)

Presented by:
Dounia Khaldi
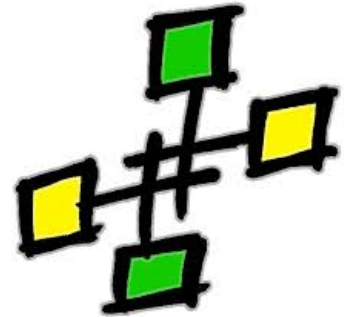
ACKNOWLEDGMENT

OUTLINE

- Motivation

- Features

- OpenUH Compiler

- Current Analyses

- How to install

- How to use (Demo)

- Conclusion

- Current Work

MOTIVATION

- Library vs. Language

- Common errors

  - Porting existing codes to OpenSHMEM

- Necessity of compile-time approach for large scale executions

- Implementation of OpenSHMEM-aware compiler within OpenUH

FEATURES

- Supported input languages: C and C++
- Source code analysis and correctness checking capabilities
- Graphical and textual display of analyses for the source program
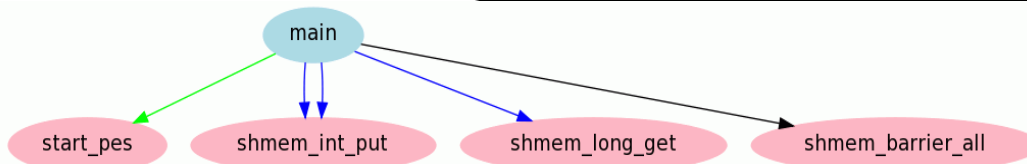- Intra-and Inter-procedural analysis

OPENUH COMPILER

- Open64-based compiler

- Sophisticated intra- and inter-procedural analyses and optimizations

- Existing support for parallel models: OpenMP, OpenACC and CAF

SHMEM ANALYZER IN OPENUH

Compilation using OpenUH
uhcc -shmem-analyzer -c

Linker (interprocedural)
uhcc –shmem-analyzer *.-o

main

start_pes   shmem_int_put   shmem_long_get   shmem_barrier_all

Legend:
Remote Memory Accesses
Reductions
Broadcast
Atomics
Memory Mgt
State Queries
Synchronizations
Contains OpenSHMEM
OpenSHMEM Call

callgraph: test-bounds

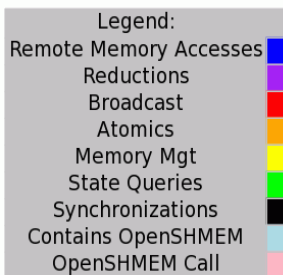| Symmetric Variable Checking |
| Bounds Checking |
| Type Checking |
| Pointer Initialization Checking |
| Barrier Matching |

Enabling parallel optimizations:
message vectorization,
communication/computation overlap...

CURRENT ANALYSES

	Collective Operations	RMA Operations	Synchronization Operations
Symmetric variable checking	X	X	X
Bounds checking	X	X	X
Type checking	X	X	X
Pointer initialization	X	X	X
Barrier matching			X

HOW TO INSTALL

- The project website for the OpenSHMEM Analyzer

  http://www.openshmem.org/OSA

- Install OpenUH (>= 3.0.38)

  http://web.cs.uh.edu/~openuh/download/

- Packages to Install:

  - Code2html
  - Graphviz

HOW TO USE OPENSHMEM ANALYZER (DEMO)

- Hints on preparing programs for OSA
- Visualization of results and manipulating graphs
- Demo videos are here

http://web.cs.uh.edu/~hpctools/openshmem

CONCLUSION

- Work in progress

- Starting point for

  - Complete semantic awareness of OpenSHMEM in the compiler

  - A comprehensive analysis and optimization framework for OpenSHMEM

CURRENT WORK

- Improving interprocedural analyses

- Extending OpenSHMEM Analyzer to PGAS using the Parallel IR of OpenUH

- More analyses such as data race detection

- Developing optimizations based on OSA