

An OpenMP Runtime Performance Analyzer

Van Bui

High Performance Computing and Tools Group
Computer Science Department

University of Houston



Problem Statement

- Learning multiple tool interfaces
- Static compiler optimization interferences
- Expensive runtime overheads
- Waiting time for post mortem performance data

Contribution

- **Novel extension to OpenMP runtime library**
 - OpenUH Compiler + PerfSuite Tool
- **Low cost method for collecting OpenMP performance data**
- **General framework for integrating performance analysis within compiler runtime environment**

Key Results

Snapshot of Statistics for a Parallel Region					
Parallel region events	Statistic	Thread 0	Thread 1	Thread 2	Thread 3
Init. overhead	Wall clock time (secs)	0	NA	NA	NA
Executing tasks	% cycles w/o instruct. issued	18.7	NV	NV	NV
	% cycles w/o instruct. completed	19.3	NV	NV	NV
	% cycles stalled on any resource	5.61	NV	NV	NV
	L1 cache miss ratio (data)	NV	NV	NV	0.11
	L2 cache miss ratio (data)	NV	NV	NV	0
	L3 cache miss ratio (data)	NV	NV	0.8	NV
	Wall clock time (secs)	0.09	0.09	0.09	0.09
Barrier wait	Wall clock time (secs)	0	NA	NA	NA

- Performance statistics for parallel regions
- Multiplexing is apparent

Weblinks

- Copper Project
 - www.cs.uh.edu/~copper
- Tools
 - OpenUH Compiler
 - www.cs.uh.edu/~openuh
 - PerfSuite Performance Tool
 - <http://perfsuite.ncsa.uiuc.edu/>