Programming and execution for DDM-CMP

Kyriakos Stavrou  Nektarios Paisios
Pedro Trancoso  Paraskevas Evripidou

University Of Cyprus
Computer Science Department
www.cs.ucy.ac.cy/carch/casper

Data Driven Multithreading (DDM)
• Program = Synchronization + Computation
• Tolerates/hides synch & comm latencies
• Implemented with commodity microprocessors
• The only additional requirement is a small hardware structure, the TSU (Thread Synchronization Unit)

Data Driven Model of execution

Program

Arbitrary collection of “serial” code and DDM Code-blocks

Producer/consumer relationship among threads

Scheduling of DDM Threads for execution is done dynamically according to data availability

Any CPU optimization used to execute the thread’s instructions

Assignment Threads to CPUs currently is statically by the compiler

The DDM-CMP C-to-C Preprocessor
• Using #pragma directives
  • #pragma ddm thread
  • #pragma ddm endthread

The DDM-CMP Compiler [Under Development]
• C-to-C compiler
  • Augments C program with ddm #pragma directives
• Compilation process
  • Find candidate program structures (e.g loops ) for DDM threads
  • Find data dependencies
  • Perform loop transformations

Program Execution [Execution on 2 CPUs of a simple program]

Experimental results
Preliminary results use extrapolation from the D2NOW system

DDM-CMP4 : CMP with 4 P3 CPUs
DDM-CMP8 : CMP with 8 P3 CPUs with reduced cache
Baseline : P4 3.2GHz

This project is funded by the Cyprus Research Promotion Foundation