



Modules in Community: Injecting parallelism throughout CS curricula

Libby Shoop

Macalester College – St. Paul, MN

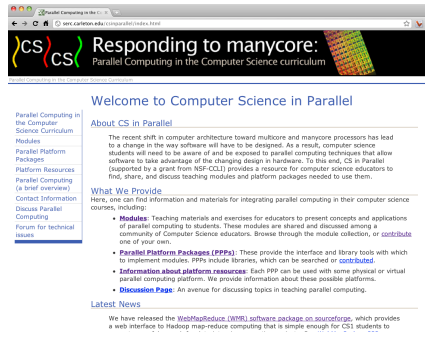
Dick Brown

Saint Olaf College – Northfield, MN



Abstract

Given the recent emergence of multi-core and distributed computing that is transforming mainstream application areas in industry, demand is rising for teaching more parallelism and concurrency in CS curricula. We argue for teaching these topics incrementally in CS courses at all undergraduate levels, and propose a comprehensive approach involving flexible teaching modules with experiential programming exercises and other supplements, support materials for parallel computing resources, and development of an online community of educators and module contributors who support each other.



Parallel Platform Packages:

A collection of *Parallel Platform Packages (PPPs)* support the modules by providing introductory documentation for standard parallel computing resources, and a small amount of software to make parallel computation more accessible.

Documentation: Although documentation exists for parallel programming libraries, Beowulf clusters, etc., these documentation PPPs provide a uniform set of starting points.

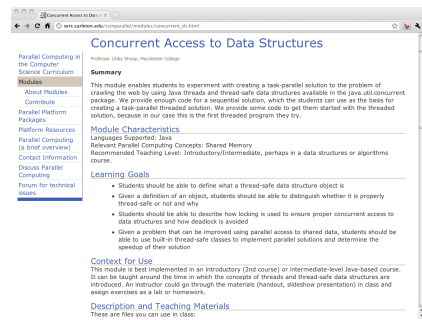
Tools: *WebMapReduce (WMR)* (see webmapreduce.sf.net/) provides strategically simplified access to the open-source Hadoop [<http://hadoop.apache.org/>] implementation of map-reduce computation, and supports an extensible range of programming languages, for hands-on exercises at all levels, including introductory courses.

A modular strategy

We believe parallel concepts should be taught early and often throughout the computer science curriculum, using effective active-learning pedagogy. In order to accomplish this, we are producing *flexible teaching modules* on topics ranging across the spectrum of parallel computing principles and practices, each designed for use in a variety course and institutional contexts. Each module contains:

- learning objectives
- comments on contexts for its use
- expository text
- student exercises, and
- assessment tools

Our goal is to make it as convenient as possible for an instructor to insert a day or two on parallel computing in a course, even when that instructor may not have taught parallelism in that course in the past.



Initial portion of a module page

Course Modules

We are developing short modules (1-3 days of class time) that are self-contained and flexible for use in various courses. Active lab modules are designed for hands-on use in class as lab activities. Conceptual modules contain readings, practice problems, class presentation slides, and homework questions. The following modules have been partially or completely produced, and represent the level and range of topics we envision.

Introductory level

Map-reduce computing using WebMapReduce

Concurrent Access to Data Structures in Java

Multicore Programming with Intel's Manycore Testing Lab

Conceptual material

Introduction to Parallel Computing Concepts

Intermediate and upper level

Parallel Sorting Algorithms

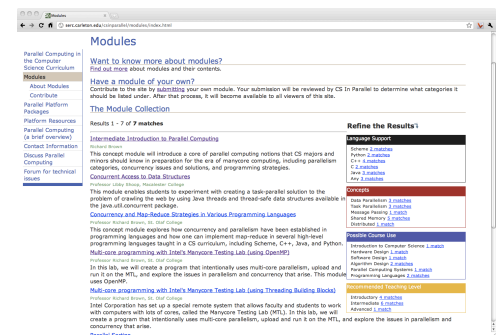
Intermediate Map-Reduce Computing

Concurrency and Map-Reduce Strategies in Programming Languages

The π -calculus (for Theory of Computation)

Growing a community of educators, csinparallel.org

We seek to develop a web-based community where instructors can obtain modules, get questions answered, share their own modules and materials, and relate experiences with the available materials, largely through peer-to-peer interaction among educators. Instructors who share material maintain their own copyright. Site features include filtering by language, course use, parallel concept, or level in curriculum, sharing comments, and “user” forums for resolving technical issues. These are offered to encourage instructors to collaborate and support each other as they incorporate more parallel computing in courses throughout the undergraduate CS curriculum



This material is based on work supported by the National Science Foundation under Grant Nos. 9041962 and 942160. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.