



## **HECToR course syllabus: “OpenMP”**

### *The Shared Memory “Machine”*

Review of memory hierarchy and cache organization; cache coherency

### *Introduction to OpenMP*

The OpenMP execution model; compiler directives, clauses, “sentinels” and pragmas; parallel regions; scoping of variables (shared, private, default); race conditions and avoiding this type of bug; lexical and dynamic extents

### *Work Sharing*

Parallelization of code; parallel do’s and for’s, parallel sections, “single” and “master” directives, “workshare”, using “nowait” for load balancing; writing reduction operations in OpenMP

### *Scheduling*

Static and dynamic scheduling of parallel operations; more on load balancing

### *Other Clauses and Directives*

Deciding if parallelization is worthwhile for a given operation; controlling the number of threads; more on scoping of variables (firstprivate, lastprivate, threadprivate); dealing with global variables. Collapsing loops.

### *Orphan Directives*

Writing OpenMP in subprograms; the behaviour of subprograms when called by each thread

### *Synchronization*

Controlling the threads to avoid errors and bugs by using barriers, critical regions and atomic operations; ensuring a consistent view of shared data with flushes; parallelization of loops with dependencies using “ordered”

## *Tasks*

Introduction and how tasks are embedded in the language from version 3.0; implicit and explicit tasks; task scoping, scheduling and synchronization.

## *Environment Variables and Run Time Library Routines*

Getting and setting information about threads, such as the number of threads or their number; using locks; timing routines

## *Hints and Optimization*

A discussion of the issues that affect the performance of OpenMP codes, including “false sharing”

## *Background Reading*

It is not essential to do any reading in advance of the course but the following references may be useful.

1. OpenMP Specification Documents, <http://openmp.org/wp/openmp-specifications/>
2. Chandra et al, “Parallel Programming in OpenMP”, Morgan Kaufmann.
3. Chapman, Jost, and van der Pas, “Using OpenMP: Portable Shared Memory Parallel Programming”, MIT Press.