DEPARTMENT CIS

AUTHOR(S) Peter Chapin

COURSE NUMBER CIS-5230

COURSE TITLE Parallel Programming

SHORT TITLE Parallel Programming

COURSE LEVEL 5000

DATE CREATED 4/2/2010

CHECKED/CHANGED 11/6/2017

PREREQUISITES CIS-2230 and CIS-3050

COREQUISITES Click here to enter corequisites (ex. **UWB 9998**, **UWB 9999**); if NA, leave blank

RESTRICTIONS Click here to enter any restrictions on enrollment; if NA, leave blank

SPECIAL FEES Click here to enter “**Yes**” or “**No**”

CREDITS 3

HOURS 3

SEMESTER Spring

COURSE DESCRIPTION This course examines the applications, algorithms, construction, configuration and performance of parallel programs. Topics include shared memory parallelism using POSIX threads and OpenMP, and multi-machine parallelism using MPI. Parallel programming on modern GPU devices is also introduced.

SUGGESTED TEXTS Introduction to Parallel Programming by Peter S. Pacheco; Morgan Kaufmann; Copyright 2011; ISBN=978-0-12-374260-5.

OPTIONAL TEXTS Click here to enter optional texts; if NA, leave blank

COURSE OUTCOMES Understand the range of parallel programming options available. Understand the issues and approaches for safely controlling concurrency. Write programs that take advantage of multiple threads in a shared memory system using the POSIX Threads API. Write programs that take advantage of directive based methods such as OpenMP. Write programs that take advantage of message passing in a multi-machine cluster using MPI. Write simple programs that take advantage of GPU based computing using CUDA and OpenACC.

COURSE CONTENT Introduction; approaches and applications for parallelism. Amdahl's Law and Flynn's Taxonomy. POSIX Thread creation and destruction. Shared memory (POSIX Thread) synchronization primitives. Parallel decomposition via recursion. Performance tradeoffs with parallelism; caching effects, thread pools. OpenMP. Lock free programming in shared memory systems. Cluster software and its configuration. MPI. Cluster network configurations; communication patterns. Parallel decomposition in clusters. GPU programming with CUDA. GPU programming with OpenACC.

LAB OUTCOMES Click here to enter lab outcomes; if NA, leave blank

LAB CONTENT Click here to enter lab content; if NA, leave blank

LECTURE CAPACITY Click here to enter capacity (standard VTC lecture cap is 32)

LAB CAPACITY Click here to enter capacity (standard VTC lab cap is 16)

GRADED OR P/NP Graded

EVALUATION Homework, Projects, Final Exam. Also: Read and summarize two papers selected by the instructor, and one paper of their own choosing (approved by the instructor). Complete a "graduate project" specified by the instructor in place of one of the regular projects that delves more deeply into the subject matter and that includes a detailed performance analysis. The student will also do a short presentation about his or her project to the class.

DELIVERY METHOD ONL

ROOM REQUIREMENTS Click here to enter room requirements (ex. **Telepresence**)

AUTHOR’S NOTES Click here to enter optional author’s notes; if NA, leave blank