

**CSC726**                      **Fall 2016**  
**Parallel Algorithms**

**Professor:** Torgersen

**Office Phone:** 758-5536

**Office Hours:** Mon. and Wed. 3:00 to 4:30, Thu. at 2:00 to 4:30 and by appointment.

**Text:** Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, Introduction to Parallel Computing.

**Facilities:** Dell PowerEdge C6145 (gottlieb), Sun 5120 (genesis), WFU Linux Cluster (deac), possibly LittleFE

**Goals:**

1. Overview of Parallel Computing Paradigms
  - (a) Speedup, Amdahl's Law and Gustafson's Law
  - (b) Fine grain vs coarse grain parallelism
  - (c) Data parallelism vs functional parallelism
  - (d) Light weight threads
  - (e) Shared memory parallel programming directives (e.g., OpenMP)
  - (f) Message passing (MPI)
2. Overview of Parallel Architecture Designs
  - (a) Pipelining and super-scalar instructions
  - (b) Shared memory and memory issues, (e.g., contention, cache coherence)
  - (c) Memory Coherency models
  - (d) Message passing and communication issues
  - (e) Traditional interconnect topologies
  - (f) Multi-core processors (e.g., SPARC T4, AMD Bulldozer)
  - (g) Vector processors (e.g., IBM G5/Altivec)
3. Theory: Parallel time complexity
  - (a) Speedup
  - (b) Scalability: Asymptotic relation between problem size and number of processors.
  - (c) Parallel time
  - (d) Efficiency: Ratio of Speedup to number of processors
4. Hands-on
  - (a) Parallel programming projects.
  - (b) Include problems from a variety of disciplines, e.g., image processing, problems from your thesis research.
  - (c) Develop skill in programming with concurrent threads.
  - (d) Program correctness: you can't verify a parallel program by testing!!

5. Some widely-studied parallel problems,
  - (a) Parallel prefix
  - (b) Matrix algorithms, e.g., QR factorization
  - (c) Sorting (e.g., bitonic sort) and Searching
  - (d) Graph algorithms
  - (e) Parallel FFT (if time allows)
6. Data dependency analysis
7. Automatic detection of parallelism (as time allows)

**Expectations:**

1. Class participation.
2. Communicate if things get complicated.
3. Your best effort.

**Grading:**

Two exams (60%), a few take home problem sets (10%), as many programming projects as we can produce (30%).

**Disability Notice:**

If you have a disability that may require an accommodation for taking this course, then please contact the Learning Assistance Center (758-5929) within the first two weeks of the semester.

**Pandemic Planning Notice:**

The University has requested that faculty collect personal contact information as part of emergency planning and preparation. The information you provide is strictly confidential.