CMPT 431 - D100 Distributed Systems

Instructor(s): Keval Vora

Calendar Objective/Description:
An introduction to distributed systems: systems consisting of multiple physical components connected over a network. Architectures of such systems, ranging from client-server to peer-to-peer. Distributed systems are analyzed via case studies of real network file systems, replicated systems, sensor networks and peer-to-peer systems. Hands-on experience designing and implementing a complex distributed system.

Instructor’s Objectives:
The course aims to provide an understanding of principles involved in designing modern parallel and distributed software systems. It focuses on the fundamentals of parallel algorithm design and parallel programming techniques by covering key concepts like concurrency, synchronization, consistency models and fault tolerance.

Prerequisites:
CMPT 300, 371. Students who have taken CMPT 401 before September 2008 may not take this course for further credit.

Topics:
- Principles of Parallel Algorithm Design
- Shared Memory Programming
- Distributed Memory Model & Programming
- Consistency Models
- Fault Tolerance
- Scalable Analytics

Grading:
There will be programming assignments, midterm, and final exam. Details about grading will be discussed in the first week of class.

Reference Books:
The Art of Multiprocessor Programming, Maurice Herlihy and Nir Shavit, Elsevier/Morgan Kaufmann, 2012, 9780123977953

Academic Honesty Statement:
Academic honesty plays a key role in our efforts to maintain a high standard of academic excellence and integrity. Students are advised that ALL acts of intellectual dishonesty will be handled in accordance with the SFU Academic Honesty and Student Conduct Policies (http://www.sfu.ca/policies/gazette/student.html).