Computer Science and Engineering CSE4163/CSE6163
Designing Parallel Algorithms

CREDIT/CONTACT HOURS: Credit Hours: 3, Contact Hours: 45

COORDINATOR:
Dr. Ed Luke

   a. Supplemental Material: No required material

SPECIFIC COURSE INFORMATION:
Catalog Description: Three hours lecture. Techniques for designing algorithms to take efficient and scalable advantage of parallel architectures. Includes techniques for parallelizing sequential algorithms and techniques for matching algorithms to architectures.
   a. Prerequisites: Grade of C or better in CSE 3324 or CSE 4733/6733
   b. Required/Elective:
      Computer Science: Selected Elective
      Software Engineering: Elective
      Computer Engineering: Elective

SPECIFIC GOALS OF THE COURSE:
a. Specific Outcomes of Instruction:
   1. The development of the students’ ability to design algorithms to run on distributed, cluster, and shared memory computing platforms
   2. The expansion of the students’ algorithm analysis experience by applying analysis to parallel and distributed algorithms.
   3. The development of the students abilities in the application of different models, both performance and programming models, to the analysis, selection, and development of distributed and parallel algorithms
b. Criterion 3 Outcomes:
   Note: Parenthesized list indicates the ABET EAC and CAC outcomes addressed by each performance criteria.
   1. The course requires the analysis of parallel algorithms using asymptotic analysis and the application of analytic performance models. In addition the course requires the student to be able to utilize these analyses to select an appropriate parallel algorithm under various processing scenarios. (EAC: a, e; CAC a,b, j)
   2. The course requires the student to develop parallel software using modern programming tools and to evaluate the performance of various implementation options. In addition, the student is expected to analyze the collected data using the theoretical models covered in item 1. (EAC: b,c,k; CAC: c, i, k)
TOPICS COVERED:

1. Introductory material – Flynn’s Taxonomy; Models of parallel computing (Shared Memory, Data Parallel, Message Passing) 3
2. Interconnection Networks – Metrics (Diameter, Connectivity, Cost, Bisection width & Bandwidth); Hypercube, mesh & Torus, Linear & Ring, Complete 2
3. Methodology of Designing Parallel Algorithms – Decomposition Techniques; Task Interaction Characteristics; Load Balancing Strategies; Mapping Techniques; Methods for Containing Overhead 6
4. Parallel Algorithm Models; Data-Parallel; Task Graph Model; Work Pool; Producer-Consumer; Hybrid 3
5. Collective Communication Algorithms; Broadcast; All-to-All; All-to-All personalized; Prefix-Sum; Reductions; Large Message Optimizations 5
6. MPI—Basic modes of communication, Basic Routines, Communicators, Collective Operations 2
7. Open-MP and shared memory programming 1
8. Quantitative Design Issues – Speedup; Efficiency, Scalability; Amdhal’s Law, Parallel Cost, Cost Optimality, Isoefficiency 6
9. Parallel Sorting – Algorithms and complexities; Bitonic; Odd-Even Transposition; Shell sort; parallel quicksort; Bucket and Sample sort 6
10. Parallel Matrix Algorithms; -- algorithms and complexities for matrix-vector and matrix-matrix operations 6
11. Exams 3