Fall 2013
Georgia Tech, CX 4140 / CSE 6140
Tu/Th 9:35am - 10:55am, Classroom: Clough Building, Room 102

Computational Science & Engineering (CSE) Algorithms
http://www.cc.gatech.edu/~bader/COURSES/GATECH/CSE-Algs-Fall2013/

Instructor: Prof. David A. Bader, KACB 1320, 404-385-0004, bader@cc
Office Hours: Tuesday 11:00am-12:00pm
Teaching Assistant 1: Lluis Miquel Munguia, lluis.munguia@gatech.edu
TA1 Office Hours: Tuesday/Thursday 11:00am-12:00pm, Klaus Room 1343
Teaching Assistant 2: Anita Zakrzewska, azakrzewska3@gatech.edu
TA2 Office Hours: Monday/Wednesday 3:00pm-4:00pm, Klaus Room 1343
Textbooks:

- Recommended: Cormen, Leiserson, Rivest, and Stein,

Course Description: This course will introduce students to designing high-performance and scalable algorithms for computational science & engineering applications. The course focuses on algorithm design, complexity analysis, experimentation, and optimization, for important science and engineering applications. Students will develop knowledge and skills concerning:

- the design and analysis of real-world algorithms employed in computational science and engineering applications, and
- performance optimization of applications using the best practices of algorithm engineering.

Pre-requisites: design and analysis of algorithms (CS 3510).
Students (from the Sciences, Engineering, and Computing) interested in algorithmic applications in science and engineering are encouraged to take this course.

This course can be taken for satisfying the theory breadth requirement by computer science graduate students (M.S. and non-theory Ph.D. students). This course cannot be taken by ACO students to satisfy their core requirement and theory Ph.D. students in computer science to satisfy the theory breadth requirement.

Grading:

(25 %) Midterm
(25 %) Final
(25 %) Project
(20 %) Homework
( 5 %) Class participation
CLASS POLICIES

1. Class announcements will be sent to the Georgia Tech T-Square mailing list, see http://t-square.gatech.edu/.

2. Please let me know as soon as possible if you will need to re-schedule an exam, or have any special needs during the semester.

3. Each student must read and abide by the Georgia Tech Academic Honor Code, see www.honor.gatech.edu.

4. Plagiarizing is defined by Webster’s as “to steal and pass off (the ideas or words of another) as one’s own: use (another’s production) without crediting the source.” If caught plagiarizing, you will be dealt with according to the GT Academic Honor Code.

5. All homework must be submitted on-time through T-Square. Homework is due by 5PM on the given due date. Late homeworks will not be accepted without a legitimate excuse and approval from the instructor.

6. When working on homework, you may work with other students in the class. However, each student must upload their own copy of the homework to T-Square with the collaborators names annotated on every copy of the submission.

7. No collaboration is permitted on exams. The midterm and final exams will be in-class, closed-book exams. You will be allowed to take a “cheat sheet” (double-sided 8.5 x 11 sheet of paper) into each exam.

8. Unauthorized use of any previous semester course materials, such as tests, quizzes, homework, projects, and any other coursework, is prohibited in this course. Using these materials will be considered a direct violation of academic policy and will be dealt with according to the GT Academic Honor Code.
Coverage of Topics

The course balances the use of theory and practice in algorithm design by presenting the student with case studies from application domains. After an introduction to computational science and engineering applications, algorithm theory, and realistic models of computation, the course consists of two modules. The first focuses on the design and analysis of real-world algorithms, with an application walk-through from algorithmic techniques to construction of the algorithm and finally application-level performance. The second module focuses on attaining high-performance implementations through algorithm engineering. These modules develop the theoretic framework and complements the study with examples from real-world problems using implementations on modern computing systems.

Computational Science & Engineering Introduction

1. Overview of Computational Sci. and Engr. Applications; characteristics and requirements
2. Computability and Complexity
3. Ideal and Realistic Models of Computation
4. Design of Parallel Algorithms
5. Performance Analysis
6. Multi-scale, multi-discipline applications

Real-World Algorithms: Techniques to Applications

1. Algorithms for Genomics: Divide and Conquer, cache-aware data structures, string algorithms
2. Sequence similarity searching: dynamic programming, string algorithms, local alignments, BLAST
3. Parallel Sorting by Regular Sampling: Randomized algorithms, load balancing, partitioning, merging, and sorting
4. Scientific visualization: greedy algorithms, 3d surface construction, marching cubes
5. Graph partitioning: NP-completeness, approximation algorithms, adaptive mesh refinement
6. Graph analysis: search algorithms, semantic networks, betweenness centrality

Algorithm Engineering

1. Modeling of the Memory Hierarchy
2. Cache-friendly, cache-aware, and cache-oblivious algorithms
3. Ideal and realistic models of parallel computation
4. Parallel and distributed algorithms and applications
5. Parallel disk models
6. Problem solving frameworks
7. Tools for Performance analysis
8. Experimental methods and validation
<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Lec</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20 Aug</td>
<td>1</td>
<td>Intro. to Computational Science &amp; Engineering</td>
</tr>
<tr>
<td></td>
<td>22 Aug</td>
<td>2</td>
<td>Modern Computer Architecture, Models of Computation, Multicore</td>
</tr>
<tr>
<td>2</td>
<td>27 Aug</td>
<td>3</td>
<td>Review of Computability and Complexity (Guest Lecturer: L. Munguia)</td>
</tr>
<tr>
<td></td>
<td>29 Aug</td>
<td>4</td>
<td>Map-Reduce Algorithms, Hadoop ; Project Discussion</td>
</tr>
<tr>
<td>3</td>
<td>3 Sep</td>
<td>5</td>
<td>Real-world algorithm analysis</td>
</tr>
<tr>
<td></td>
<td>5 Sep</td>
<td>6</td>
<td>Amdahl’s Law, Memory Hierarchy, Memory Wall, Principle of locality, Cache design</td>
</tr>
<tr>
<td>4</td>
<td>10 Sep</td>
<td>7</td>
<td>String and tree algorithms (Guest Lecturer: L. Munguia)</td>
</tr>
<tr>
<td></td>
<td>12 Sep</td>
<td>8</td>
<td>Biological Sequence Alignment and Searching (Guest Lecturer: L. Munguia)</td>
</tr>
<tr>
<td>5</td>
<td>17 Sep</td>
<td>9</td>
<td>Practical Graph Algorithms</td>
</tr>
<tr>
<td></td>
<td>19 Sep</td>
<td>10</td>
<td>Performance Analysis, Multicore and High-Performance Computing</td>
</tr>
<tr>
<td>6</td>
<td>24 Sep</td>
<td>11</td>
<td>Multi-scale, Multi-physics applications</td>
</tr>
<tr>
<td></td>
<td>26 Sep</td>
<td>12</td>
<td>Classes: P, NP, Reductions, NP-completeness proofs</td>
</tr>
<tr>
<td>7</td>
<td>1 Oct</td>
<td>13</td>
<td>Large-scale Graph and Multithreaded Algorithms</td>
</tr>
<tr>
<td></td>
<td>3 Oct</td>
<td>14</td>
<td>Graph Partitioning (with Nodal Coordinates)</td>
</tr>
<tr>
<td>8</td>
<td>8 Oct</td>
<td>15</td>
<td>Graph Partitioning (Spectral Methods)</td>
</tr>
<tr>
<td></td>
<td>10 Oct</td>
<td>16</td>
<td>Multi-Grid Methods</td>
</tr>
<tr>
<td>9</td>
<td>15 Oct</td>
<td>-</td>
<td><strong>Fall Break</strong></td>
</tr>
<tr>
<td></td>
<td>17 Oct</td>
<td>17</td>
<td>Midterm</td>
</tr>
<tr>
<td>10</td>
<td>22 Oct</td>
<td>18</td>
<td>Streaming Graph Analytics, Part 1 (Guest Lecturer: A. Zakrzewska)</td>
</tr>
<tr>
<td></td>
<td>24 Oct</td>
<td>19</td>
<td>Streaming Graph Analytics, Part 2 (Guest Lecturer: A. Zakrzewska)</td>
</tr>
<tr>
<td>11</td>
<td>29 Oct</td>
<td>20</td>
<td>Real-world Connectivity Algorithms (Routing, Distance)</td>
</tr>
<tr>
<td></td>
<td>31 Oct</td>
<td>21</td>
<td>Real-world Centrality Algorithms (Informatics and Interactions)</td>
</tr>
<tr>
<td>12</td>
<td>5 Nov</td>
<td>22</td>
<td>Realistic models of parallel computation</td>
</tr>
<tr>
<td></td>
<td>7 Nov</td>
<td>23</td>
<td>Vectorized Algorithms and Techniques (SIMD, SSE intrinsics, GPU, CUDA)</td>
</tr>
<tr>
<td>13</td>
<td>12 Nov</td>
<td>24</td>
<td>Accelerator (GPU) Algorithms</td>
</tr>
<tr>
<td></td>
<td>14 Nov</td>
<td>25</td>
<td>Experimental Methods and Validation</td>
</tr>
<tr>
<td>14</td>
<td>19 Nov</td>
<td>26</td>
<td>Project Presentations, Session 1</td>
</tr>
<tr>
<td></td>
<td>21 Nov</td>
<td>27</td>
<td>Project Presentations, Session 2</td>
</tr>
<tr>
<td>15</td>
<td>26 Nov</td>
<td>28</td>
<td>Algorithm Engineering and Performance</td>
</tr>
<tr>
<td></td>
<td>28 Nov</td>
<td>-</td>
<td><strong>Thanksgiving Break</strong></td>
</tr>
<tr>
<td>16</td>
<td>3 Dec</td>
<td>29</td>
<td>Problem Solving Environments for Discrete Algorithms</td>
</tr>
<tr>
<td></td>
<td>5 Dec</td>
<td>30</td>
<td>Special Topics: Open Problems in CSE Algorithms</td>
</tr>
<tr>
<td>17</td>
<td>12 Dec</td>
<td>-</td>
<td><strong>Final Exam</strong> (8:00am-10:50am)</td>
</tr>
</tbody>
</table>
Additional Readings:


