CS 475/575 – Introduction to Parallel Programming

Catalog Description: Theoretical and practical survey of parallel processing, including a discussion of parallel architectures, parallel programming paradigms, and parallel algorithms. Programming one or more parallel computers in a high-level parallel language.

Credits: This course combines approximately 90 hours of instruction, and approximately 30 hours of programming work for 4 credits.

Terms Offered: Spring (in even-numbered years)

Prerequisites: CS 325

Plus, this course will be a very intense experience in C/C++ programming. As such, you should come in already proficient in, at least, C. This will not be a good time to learn C from scratch. You should also be familiar with data structures. Some knowledge about computer architecture (e.g., cores, cache) would be a plus.

Courses that require this as a prerequisite: None

Structure: Three 50-minute lectures per week
Note: Due to extensive programming assignments, this course has an implied, non-scheduled lab. The lab takes place on the students’ own times, and is not part of the official course schedule. TAs are available to help the students with programming assignments at times announced in the syllabus.

Instructor: Mike Bailey

Measurable Student Learning Outcomes for both CS 475 and CS 575 students:
On completion of the course, students will have demonstrated the ability to:

1. Explain the clock speed limitations of computing, using physics and Moore's Law (ABET Outcomes a, j)
2. Explain the limitations of parallel computing using Amdahl's Law (ABET Outcomes a, j)
3. Demonstrate “parallel thinking” in program design (ABET Outcome b)
4. Explain the difference between ILP, TLP, DLP, and SIMD and the advantages of each (ABET Outcomes b, j)
5. Demonstrate the ability to program parallel algorithms in TLP, DLP, and SIMD. (ABET Outcomes b, k)
6. **Analyze** and **Characterize** different parallel programming patterns and what types of problems they best address (ABET Outcomes b, c, j, k)
7. **Characterize** how cache issues affect parallel performance (ABET Outcomes b, j)
8. **Demonstrate** the proper use of synchronization to avoid race conditions and deadlock (ABET Outcomes b, c, j)
9. **Characterize** the advantages and disadvantages of using a GPU versus using a

**Additional Measurable Student Learning Outcome for CS 575 Students:**

10. **Read** a current research paper in parallel computing and write a 5-page paper **Analyzing** it. (ABET Outcomes a, b, c, f, g, l, j)

**Learning Resources:**

**Course Content:**
- Parallel computing: types, limitations
- Moore’s Law, Amdahl’s Law
- SIMD
- OpenMP
- Synchronization issues in parallel computing
- Cache issues in parallel computing
- Threading Building Blocks
- OpenCL
- MPI

**Evaluation of Student Performance:**
Projects: 800 points
Tests (2) 200 points total (i.e., 100 points each)
TOTAL 1000 points

<table>
<thead>
<tr>
<th>Points</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>950</td>
<td>A</td>
</tr>
<tr>
<td>925</td>
<td>A-</td>
</tr>
<tr>
<td>875</td>
<td>B+</td>
</tr>
<tr>
<td>850</td>
<td>B</td>
</tr>
<tr>
<td>825</td>
<td>B-</td>
</tr>
<tr>
<td>775</td>
<td>C+</td>
</tr>
<tr>
<td>750</td>
<td>C</td>
</tr>
<tr>
<td>725</td>
<td>C-</td>
</tr>
<tr>
<td>675</td>
<td>D+</td>
</tr>
<tr>
<td>650</td>
<td>D</td>
</tr>
<tr>
<td>625</td>
<td>D-</td>
</tr>
</tbody>
</table>
Students with Disabilities:
Accommodations are collaborative efforts between students, faculty and Disability Access Services (DAS). Students with accommodations approved through DAS are responsible for contacting the faculty member in charge of the course prior to or during the first week of the term to discuss accommodations. Students who believe they are eligible for accommodations but who have not yet obtained approval through DAS should contact DAS immediately at 737-4098.

Link to Statement of Expectations for Student Conduct, i.e., cheating policies
http://oregonstate.edu/studentconduct/code/index.php

Wordle from the Course Notes: