CS 798: Multicore programming—Fall 2019

Instructor	Trevor Brown
Office hours / Tutorials	In synchronous online call, time TBD by Doodle poll
E-mail	trevor (dot) brown (at) uwaterloo (dot) ca
Times	Asynchronous pre-recorded videos — watch when you like

Overview. Want to learn what it takes to efficiently harness dozens or hundreds of cores in a multicore system? In such a system, the difference between the performance of a bad implementation and a good one can easily be 100x. This course uses state of the art concurrent data structures as examples to explain high performance implementation techniques and surprising performance pitfalls. Students should finish the course with knowledge of several concurrent data structures, and an understanding of how data structures interact with various aspects of real systems, including thread scheduling, memory allocation, processor caches, non-uniform memory and multi-socket architectures.

Classes. Two thirds of the course will consist of lectures on the following topics.

- Asynchronous shared memory, counters, linearizability, cache coherence, false sharing
- Lock-freedom, lock-free stack
- Relaxed ordering and bags, breadth first search
- Unordered maps: hashing and hash tables
- Limits of parallelism in problems, data parallelism, task dependencies, OpenMP
- Multi-word synchronization primitives such as multi-word compare-and-swap
- (Hardware) transactional memory
- Ordered maps: binary search trees, correctly implementing atomic updates, searches without locking
- Memory reclamation with and without locks, epoch based reclamation

Presentations. Towards the end of the course, each student will choose a paper to read and give a conference-style presentation on (format to be determined—possibly recorded video or live MS Teams/Zoom call). Following his/her presentation, each student will answer questions about the paper in an online forum (participation marks for students who ask good questions). On student presentation days, students *not* presenting will read one of the papers being presented, and submit a short review on it. In a week where you are presenting, you do not need to submit any reviews.

The following would make good presentation topics. Of course, I will recommend papers on these topics (and can suggest papers for other topics by request).

- Synchronization mechanisms: read-copy-update, read-log-update, flat combining, lock-elision, transactional memory (software, hardware and hybrid)
- Scalable multi-threaded memory allocation, NUMA-aware memory allocation
- Other memory reclamation algorithms such as hazard pointers
- Performance impact of non-uniform memory architectures (NUMAs), NUMA-aware data structures
- Relaxed memory consistency models, instruction reordering
- Non-volatile RAM (NVRAM) / persistent memory / Intel Optane DCPIMM

Assignments. Assignments will mostly involve programming in C++. There will also be some theoretical questions, especially surrounding *correctness* of concurrent data structures.

Assignments will be submitted on *Marmoset* (programming parts) and *Crowdmark* (written parts). When you submit on Marmoset, your algorithm will be run on a multicore machine, and performance results will be shown to you (with a baseline implemented by me or a TA shown as a comparison). As assignments are in C++, some background in C or C++ is recommended.

Evaluation. Grades will be calculated as follows:

20%	Paper presentation
10%	Reviews of papers
10%	Participation in synchronous meetings and online discussion forum
60%	Assignments

Course classification. Note that this course can count for either HW/SYS or AC. That is, once the course is done, you can choose for it to count as *either* HW/SYS or AC (but not both) towards your breadth requirements. This makes the course fairly flexible re: requirements...

Academic Integrity. Note that students are not generally permitted to submit the same work for credit in multiple classes. For example, if a student has reviewed or presented one of the papers in another seminar class, he or she should avoid reviewing or presenting it again for this class. The general university policy:

- Academic Integrity: In order to maintain a culture of academic integrity, members of the University of Waterloo community are expected to promote honesty, trust, fairness, respect and responsibility. Check the Office of Academic Integrity's website for more information.

  All members of the UW community are expected to hold to the highest standard of academic integrity in their studies, teaching, and research. This site explains why academic integrity is important and how students can avoid academic misconduct. It also identifies resources available on campus for students and faculty to help achieve academic integrity in — and out — of the classroom.

- Grievance: A student who believes that a decision affecting some aspect of his/her university life has been unfair or unreasonable may have grounds for initiating a grievance. Read Policy 70 — Student Petitions and Grievances, Section 4. When in doubt please be certain to contact the department's administrative assistant who will provide further assistance.

- Discipline: A student is expected to know what constitutes academic integrity, to avoid committing academic offenses, and to take responsibility for his/her actions. A student who is unsure whether an action constitutes an offense, or who needs help in learning how to avoid offenses (e.g., plagiarism, cheating) or about "rules" for group work/collaboration should seek guidance from the course professor, academic advisor, or the Undergraduate Associate Dean. For information on categories of offenses and types of penalties, students should refer to Policy 71 — Student Discipline. For typical penalties, check Guidelines for the Assessment of Penalties.

- Avoiding Academic Offenses Most students are unaware of the line between acceptable and unacceptable academic behaviour, especially when discussing assignments with classmates and using the work of other students. For information on commonly misunderstood academic offenses and how to avoid them, students should refer to the Faculty of Mathematics Cheating and Student Academic Discipline Policy.

- Appeals: A decision made or a penalty imposed under Policy 70, Student Petitions and Grievances (other than a petition) or Policy 71, Student Discipline may be appealed if there is a ground. A student who believes he/she has a ground for an appeal should refer to Policy 72 — Student Appeals.

Students with Accessibility Needs. AccessAbility Services, located in Needles Hall, Room 1401, collaborates with all academic departments to arrange appropriate accommodations for students with disabilities without compromising the academic integrity of the curriculum. If you require academic accommodations to lessen the impact of your disability, please register with AccessAbility at the beginning of each academic term.