COMP-512: Distributed Systems

Bettina Kemme, Joseph D'silva

kemme@cs.mcgill.ca, joseph.dsilva@cs.mcgill.ca

Names and Numbers

- ❑ Lecture:
  - ☆ Tuesday/Thursday, 1:05-2:25
  - ☆ Zoom lin: https://mcgill.zoom.us/j/96972782081
  - ☆ Hybrid mode:
    - Some lectures recorded to be watched before class time; class time then used for solving example problems interactively and for questions
    - Many lectures will be live and then recorded
  - ☆ Two instructors: how will it work?
- ❑ Office Hours:
  - ☆ After class – subject to available time.

    (public; same zoomlink as lecture)

  Private office hours:

  - ☆ Bettina Kemme: Wednesday 10am (might change): https://mcgill.zoom.us/j/97803411295
  - ☆ Joseph D'silva: Monday 11am

    https://mcgill.zoom.us/j/94865924800
- ❑ Teaching Assistants: Jianhao Cao (project), Ting Gu, David Echomgbe

Marking Scheme

- Marking Scheme:
  - ☆ 4 best of 5 Quizzes (10%)
  - ☆ 2 Assignments (total of 10%, 5% each)
    - Typically due within 10-15 days
  - ☆ 1 ungraded practice assignment (0%)
  - ☆ Programming Project(s) (30%)
    - Each delivery 10%
    - Typically due within 2 weeks
  - ☆ Midterm (15%)
    - Will be be sometime end of October; exact date to be determined in the next weeks.
  - ☆ Final (35%)
  - ☆ In the event of extraordinary circumstances beyond the University's control, the evaluation scheme of the course is subject to change, provided that there are timely communications to the students regarding the change.

Written Assignments

- ❑ Solving concrete problems, from easy to difficult
  - ☆ label an execution with timestamps according to some existing timestamping algorithm
  - ☆ extend an existing algorithm / write the extensions down in pseudo-code
  - ☆ discuss the pro/cons of two different algorithms that solve the same problem
- ❑ Usually to be done within two weeks.
- ❑ Late turn-in:
  - ☆ Will be announced when assignment is handed out
- ❑ group assignments with groups of 2
  - ☆ Random group for each assignment

Project

- ❑ Given: a simple client/server application
- ❑ Goal:
  1. distribute application
  2. make application transactional
  3. Tentative: make it fault-tolerant OR different project on Zookeeper
- ❑ Experiment with various communication paradigms
- ❑ Implement fundamental distributed algorithms
- ❑ Programming Language: Java
- ❑ In groups of three
- ❑ Find your own teammates by using the discussion forum in MyCourses
  - ☆ Discussions → Projects → Finding a group partner
  - ☆ Due to the reduced opportunities for incoming graduate students to interact with each other, we recommend that you take this opportunity to make teams within yourselves (not mandatory).

Academic Integrity

- McGill University values academic integrity. Therefore all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures (see http://www.mcgill.ca/students/srr/honest/ for more information).

French / English

- ❑ In accord with McGill University's Charter of Students' Rights, students in this course have the right to submit in English or in French any written work that is to be graded.

Literature

- ❑ Mycourses
  - ☆ All lecture slides will be made available on myCourses
  - ☆ (Life) Lecture recordings will be available on myCourses
- ❑ Books:
  - ☆ G. Coulouris, J. Dollimore, T. Kindberg, Distributed Systems: Concept and Design
    - 4th or 5th edition, Addison-Wesley

Course objectives

- ❑ Understand ***general problems***
- ❑ ***Building blocks of solutions (algorithms and implementations)***
- ❑ Develop ***the skills*** and know-how necessary to develop distributed algorithms and design distributed systems
- ❑ Understand ***trade-offs***
  - ☆ ***Performance***
  - ☆ ***Correctness***
  - ☆ ***Data consistency***
  - ☆ ***Fault-tolerance***
- ❑ Provide an overview over some ***state-of-the-art*** developments in distributed systems (research and practice)

Outline I

- Communication alternatives:
  - Interprocess communication
  - Client/server communication
    - Remote Method Invocation
  - Multicast
- Basics in distributed systems
  - Logical time
  - Coordination

Outline II

- ❑ Distributed Transactions
  - ☆ Basics -- Data and Operations
  - ☆ Concurrency Control
  - ☆ Atomic Commit Protocols
- ❑ Data Replication for Fault-Tolerance
  - ☆ Paxos
  - ☆ Zookeeper
- ❑ *Bitcoin/Blockchain*
- ❑ Large scale data management
  - ☆ Distributed File Systems
  - ☆ Distributed data management (key-value stores)
  - ☆ Distributed data processing (map-reduce, Spark, ML architectures)
- ❑ Not covered:
  - ☆ Mobile technology

Performance Implications

Algorithms

Systems Aspects

Other courses

❑ Winter 2021
  - ☆ Concurrent Programming
  - ☆ Computer Networks