Course Outline

Professor: Dr. Dorina Petriu  
Office hours: Thursdays, 4:00 - 5:30 pm
Room 5205 Canal Building
Tel: (613) 520-2600 x5652
Email: petriu (at) sce (dot) carleton (dot) ca

Calendar Description:
Designing software to demanding performance specifications. Design analysis using models of computation, workload, and performance. Principles to govern design improvement for sequential, concurrent and parallel execution, based on resource architecture and quantitative analysis.
Prerequisite(s): SYSC 5704 (ELG 6174) and a course in software engineering, or equivalent.

Course Description and Objective:
Performance in this course deals with time and capacity characteristics of computer-based systems. When systems are designed for functionality only, the time to complete a response may be so long that the system is ineffective; similarly, its capacity to serve a large number of users may be inadequate making the system uneconomic to use. Both of these problems occur often in practice. Performance engineering that aims at solving these problems is a body of concepts and techniques for evaluating systems and system designs, using measurements and models.
Meeting performance requirements (such as response time, throughput, etc.) is a major concern for all kinds of software products with performance constraints, and especially for real-time systems. Software Performance Engineering (SPE) addresses performance issues throughout the whole software lifecycle and aims to ensure that software products under development will meet their performance requirements. SPE uses predictive performance models to assess different design alternatives at an early stage, before major obstacles to performance are frozen in design and code. This can improve the quality of the final product by helping designers to make informed choices and trade-offs early in the life cycle, when changes are not expensive and open alternatives still exist. As the product evolves, so does the performance model, capturing more system features and producing more accurate results.

The course will cover different basic approaches to performance engineering. Topics will be chosen from measurement techniques, interpreting and comparing results, models that explain capacity constraints and delays (bottleneck models, queueing models), an introduction to hard-real-time delay constraints and schedulability evaluation for embedded systems and software performance engineering.

The goal of this course is to prepare the students to address performance problems in real-time concurrent and distributed systems, such as embedded controllers, enterprise distributed systems, web services-based systems and cloud systems. It will introduce the conceptual framework and the nature of performance problems and solutions, so that the student can go into the field.
The main objective/outcome of this course is to teach the students how to use performance engineering tools for design, experimentation, simulation, visualization, and analysis.

Prerequisites
Students who have not satisfied the prerequisites for this course must: a) withdraw from the course; b) obtain a prerequisite waiver or c) may be deregistered from the course after the late registration deadline.
Lectures:
When: Wednesdays and Fridays, 4:00 - 5:30 pm
Where: ME 4499

Laboratory Sessions:
When: Mondays, 2:30 - 5:30 pm (alternate weeks)
Where: AA 508

Textbook:
The following textbook will be the primary reference:
Edward D. Lazowska, John Zahorjan, G. Scott Graham, Kenneth C. Sevcik, 
“Quantitative System Performance Computer System Analysis Using Queueing 
Network Models”, Prentice Hall.
(free download from http://www.cs.washington.edu/homes/lazowska/qsp/).

References:
Daniel A. Menasce, Lawrence W. Dowdy, and Virgilio A.F. Almeida. “Performance by 

Marking Scheme:
20% Assignments
35% Project
10% Project Presentations
35% Final Exam (Centrally scheduled, 3 hours).

The final exam will be held during the formal examination period set out in the University 
Calendar and will be scheduled by Exam Services. The instructor cannot accommodate any 
special requests or alternate arrangements. The final exam is for evaluation purposes only and 
will not be returned to the student. Students who miss the final exam may be granted permission 
to write a deferred examination (see the Graduate Calendar for regulations on deferred exams).

Assignments:
Students are encouraged to discuss issues when working on assignments; however, you are 
expected to submit your own work for grading (assignments are individual work). There is a fine 
line between cooperating with your colleagues (discussing problems and ideas) and copying 
solutions (plagiarism). Not only plagiarism is an instructional offence (see the Undergraduate 
Calendar), but doing the assigned work by yourself is by far the best way to prepare for the 
exams.
Submission: Assignments are due at midnight of the due date and must be submitted online on 
cuLearn. When submitting assignments, double check that your material has indeed been 
submitted.
Late assignments will be graded according to the following policy: a cumulative 10% penalty per 
day (i.e., 24 hours) with a maximum of two days.

Project:
An important part of the course will be a project consisting of an in-depth performance analysis 
of a system or a research issue. The project will be done in teams of two students, that will be
constituted at the beginning of the term. Instructions for project proposals will be posted in the first weeks of the term.

**Copyright:**

Classroom teaching and learning activities, including lectures, discussions, presentations, etc., by both instructors and students, are copy protected and remain the intellectual property of their respective author(s). All course materials, including PowerPoint presentations, outlines, and other materials, are also protected by copyright and remain the intellectual property of their respective author(s).

Students registered in the course may take notes and make copies of course materials for their own educational use only. Students are not permitted to reproduce or distribute lecture notes and course materials publicly for commercial or non-commercial purposes without express written consent from the copyright holder(s).

**Academic Accommodation**

You may need special arrangements to meet your academic obligations during the term. For an accommodation request, the processes are as follows:

**Pregnancy obligation:** write to me with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details see the Student Guide (https://carleton.ca/equity/?p=191).

**Religious obligation:** write to me with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details see the Student Guide.

**Academic Accommodations for Students with Disabilities:** The Paul Menton Centre for Students with Disabilities (PMC) provides services to students with Learning Disabilities (LD), psychiatric/mental health disabilities, Attention Deficit Hyperactivity Disorder (ADHD), Autism Spectrum Disorders (ASD), chronic medical conditions, and impairments in mobility, hearing, and vision. If you have a disability requiring academic accommodations in this course, please contact PMC at 613-520-6608 or pmc@carleton.ca for a formal evaluation. If you are already registered with the PMC, contact your PMC coordinator to send me your Letter of Accommodation at the beginning of the term, and no later than two weeks before the first in-class scheduled test or exam requiring accommodation (if applicable). After requesting accommodation from PMC, meet with me to ensure accommodation arrangements are made. Please consult the PMC website for the deadline to request accommodations for the formally-scheduled exam (if applicable).

**Health and safety:**


**Plagiarism:**

Plagiarism (copying and handing in for credit someone else's work) is a serious instructional offense that will not be tolerated. Please refer to the section on instructional offenses in the Undergraduate Calendar for additional information.

**Tentative Week-by-Week Outline**

The following is a tentative outline of the course; it might change, based on time constraints:

Week 1: Performance concepts and requirements.
Week 2: Performance measurement. Workloads.
Week 4: Memory hierarchy effects.
Week 5: Queueing Analysis.
Week 6: Software resources.
Winter Break. Classes are suspended.
Week 7: Layered resource effects.
Week 8: Measurement and tools.
Week 9: Hard real-time systems. Schedulability analysis for hard real-time systems
Week 10: Software Performance Engineering.
Week 11: Software Performance Engineering (continued)
Week 12: Case Study.