COMP 621 Suggested Special Topic Lectures - Fall 2015

Due Dates:

- Signup by Oct 14th,
- Oct 19, 1 presentation
- Oct 21, 1 presentation
- Oct 26, 1 presentation
- Oct 28, 1 presentation
- Nov 2, 1 presentation
- Nov 4, 1 presentation
- Nov 9, 1 presentation
- Nov 11, 1 presentation
- Nov 16, 1 presentation
- Nov 18, 2 presentations

1 Introduction

The purpose of the special topics lectures is to expand the topics covered in the class and to give students an experience presenting the topic to the class. The slides and example questions will be made available via the google spreadsheet.

The special topics will be given out on a first-come, first-served basis. You may sign-up for your selected topic by entering your name, title of the presentation, and link to the paper on the google spreadsheet, the link will be in your class e-mail.

You are also free to choose your own special topic. If you wish to select your own topic, please contact the instructor to discuss the topic and the selection of research papers used for that topic.

Please sign up by October 14th.

2 Format

Each presentation should be around 40 minutes, with 5 minutes for questions. On days in which there is one special topic, the remaining class time will be an ordinary lecture.

You should prepare a very clear presentation which is intended to teach the key concepts about your special topic. You should prepare slides which: (1) motivate the area; (2) introduce the idea at a high level; (3) provide a more in depth presentation of at least a part of topic; and (4) conclude.

Your presentation should include: (1) 2 short answer questions, to be answered by the class during or at the end of your presentation; and (2) a longer question that should take between 20-30 minutes to solve. This last question will be provided as one possible question to answer on assignment #3.

3 Suggested Special Topics

Please note that papers from the ACM Digitial Library are freely available from a McGill IP, so access via a McGill network or VPN.

PLDI high-impact papers: Each year PLDI gives an award for the highest impact paper from 10 years earlier. Many of these papers are classics in compiler research. You may choose any one of them from `http://www.sigplan.org/Awards/Conferences/PLDI/Main`.

Using BDDs for pointer analysis:
`http://www.sable.mcgill.ca/bdd/`

Program Slicing:
`http://dl.acm.org/citation.cfm?id=869354`

Clone Detection:
`http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.2794`
and/or
`http://research.cs.queensu.ca/TechReports/Reports/2007-541.pdf`.

Java Decompilation:
The paper or papers listed on `http://www.sable.mcgill.ca/dava/`.

Machine Learning and Inlining:
`http://www.anc.ed.ac.uk/machine-learning/colo/inlining.pdf`.

Compiling for GPUs: Some selection of materials available at:
`http://www.pgroup.com/resources/articles.htm#papers`.

On stack replacement:
`http://dl.acm.org/citation.cfm?id=776288`.

Online optimizations:
`http://dl.acm.org/citation.cfm?id=1133981.1134010`.

MATLAB specializing JIT:
`http://www.sable.mcgill.ca/mclab/mcvm_mcjit.html`.

Auto-parallelization and machine learning:
`http://dl.acm.org/citation.cfm?id=1542496`.

The eval that men do:
`http://dl.acm.org/citation.cfm?id=2032503` and/or `http://cs.au.dk/~amoeller/papers/unevalizer/`

Retargeting Android applications to Java bytecode:
`http://dl.acm.org/citation.cfm?id=2393596.2393600`

Adaptive multi-level compilation in a trace-based Java JIT compiler:
`http://dl.acm.org/citation.cfm?id=2384616.2384630`

The impact of optional type information on JIT compilation of dynamically typed languages:
`http://dl.acm.org/citation.cfm?id=2047849.2047853`

Specifying and implementing refactorings:
`http://dl.acm.org/citation.cfm?id=1869459.1869485`

Statistically rigourous Java performance evaluation:
http://dl.acm.org/citation.cfm?id=1297027.1297033

Kind Analysis for Matlab:
http://dl.acm.org/citation.cfm?id=2048077

Fortran back-end for Matlab and shape analysis (Mc2For):
http://www.sable.mcgill.ca/mclab/projects/mc2for/.

X10 back-end for Matlab and integerokay analysis (MiX10):
http://www.sable.mcgill.ca/mclab/projects/mix10/.

Performance of JavaScript for Numerical Programs:
http://www.sable.mcgill.ca/mclab/projects/ostrich/ and http://dl.acm.org/citation.cfm?id=2661090.