

Topics in Runtime Systems – Fall 2004

University of Massachusetts, Amherst

www.cs.umass.edu/~emery/cmpsci691s

Instructor Information

Professor Emery Berger (emery@cs.umass.edu)

Office: CMPS 344

Office hours by appointment only

Course Description

Runtime systems are the glue that binds programming languages, operating systems and architecture. Topics to be covered include: explicit and automatic memory management (garbage collection), thread management and synchronization, and dynamic, just-in-time compilation. The goal of this class is to familiarize students with the key ideas behind these topics. We will be reading and discussing seminal papers as well as more recent advances in the state-of-the-art.

Textbook

Garbage Collection, by Jones & Lins. This is an optional text, but I highly recommend it.

Course Grade

Your course grade will be based on class participation, including **paper summaries prepared before each class** (20%), your presentations (20%), and a course project (60%).

Participation

Prior to each class, you should read and submit a summary/review of each paper. We will follow a conference-style review format modeled on that used at programming language conferences like PLDI, OOPSLA, and ASPLOS. Unlike most papers one reviews for a conference, though, these papers are seminal contributions. Accordingly, you should focus your review on the key ideas of the paper, and explain why they are significant. I expect you to spend approximately one hour reading each paper, unless you are the presenter. **NOTE:** Unlike real reviews, your review should only be about two paragraphs long.

Everyone will sign up to present papers to the class and lead the discussion. The role of the presenter is to explain the key ideas to everyone in the class. You should be prepared to answer questions, and so you should read the paper in detail.

To make things interesting, half of the class will be designated *advocates* and half of the class will be *detractors*. No one may remain neutral (except me!). We will swap who is an advocate and who is a detractor for each class, although the paper presenter will always be an advocate. Depending on the size and composition of the class, we may also have panel discussions and debates.

Late Policy

Just as with conferences, late submissions are **not accepted**. However, unlike conferences, I will accept a doctor's note or similar proof of a valid reason for lateness.

Topics

Topics to be covered include the following:

Virtual machines, dynamic optimization & just-in-time compilation

Synchronization

- Atomic operations

- Thin locks

- Lock-free algorithms

Thread scheduling (Cilk, Hood)

Improving Locality

- Cache-oblivious computation

- GC and locality

Dynamic storage allocation

- Hoard

- Heap Layers

- Reap

Garbage collection, including:

- Conservative GC

- Generational garbage collection

- Mark-copy (Cheney)

- Deferred reference counting (Deutsch)

- Real-time collection (Baker's Treadmill)

- On-the-fly collection (Dijkstra, Steele)

- Multiprocessor collection (Lamport)

- MP compaction (Steele)



Class Schedule

This schedule is subject to change, but we should be adhering pretty closely to it. Any changes will be announced in class and/or by e-mail.

1. Th September 9 Introduction to Runtime Systems
Emery Berger
2. Tu September 14 Adaptive Optimization in the Jalapeño JVM
Matthew Hertz
Dynamo: A Transparent Dyn. Optimization System
Trek Palmer
3. Th September 16 Thin Locks: Featherweight Synchronization for Java
Tim Richards
Wait-Free Synchronization
Michael Bradshaw
4. Tu September 21 Cilk: An Efficient Multithreaded Runtime System
Trevor Strohman
Thread Scheduling for Multi. Multiprocessors
Tom Heydt-Benjamin
- Th September 23 *class cancelled*
5. Tu September 28 Cache-Oblivious Algorithms
Bruno Ribeiro
The Garbage Collection Advantage: Improving Program Locality
Ting Yang
6. Th September 30 Hoard: A Scalable Memory Allocator
Composing High-Performance Memory Allocators
Emery Berger (both)
7. Tu October 5 Reconsidering Custom Memory Allocation
Emery Berger
Garbage Collection in an Uncooperative Environment
Gene Novark
8. Th October 7 A Non-Recursive List Compacting Algorithm
Zongfang Lin
List Processing in Real-Time on a Serial Computer
Ed Walters
- Tu October 12 *class cancelled (ASPLOS)*

Th October 14	<i>class cancelled (ASPLOS)</i>
9. Tu October 19	Generational Scavenging <i>Ting Yang</i> Simple Generational GC and Fast Allocation <i>Eric Feng</i>
10. Th October 21	An Efficient Incremental Automatic GC <i>Pritesh Sharma</i> Ulterior Reference Counting <i>Gene Novark</i>
Tu October 26	<i>class cancelled (OOPSLA)</i>
Th October 28	<i>class cancelled (OOPSLA)</i>

