Resource Virtualization

Syllabus

Web Page

http://www.cs.northwestern.edu/~pdinda/virt

Instructor

Peter A. Dinda
Technological Institute, Room L463
847-467-7859
pdinda@northwestern.edu
Office hours: Thursdays, 2-3:30pm or by appointment

Teaching assistant

Jack Lange
Ford 2-221
847-467-4708
jarusl@northwestern.edu
Office hours: Wednesdays and Fridays, 10:30-12

Location and Time

Lecture: Tech M152 (we will probably move this to Ford S-340)
Tuesdays and Thursdays, 3:30-5

TA-led Recitation/Discussion: Computer Systems Lab (Ford S-340)
Mondays, 5-6pm

Prerequisites

Required	CS 213 or (ECE 205 and ECE 231) or equivalent computer systems course
Highly recommended	CS 343 or equivalent operating systems course
Highly recommended	CS 340 or equivalent networking course
Highly recommended	Familiarity with a systems programming language such as C or C++. Familiarity with a scripting language such as Perl or Python.
Recommended	Familiarity with computer architecture, ideally to the level of ECE 361. (CS 213 is the minimum sufficient level)

The purpose behind these prerequisites is to assure that you understand the principles of how processors, computer systems, and networks work coming into the class and are able to do systems-level programming. *If you do not meet some of these prerequisites, but feel you are prepared to take the course, please contact me.*

Readings

Readings for the course will be in the form of research papers. A separate reading list is provided. There is no textbook, although I will place a copy of Smith and Nair's excellent new book on reserve in the library.

Objectives, framework, philosophy, and caveats

A basic principle in computer science is that of indirection, creating a new layer between two existing layers. By introducing a new layer into a software system, it often becomes straightforward to do many things and the system becomes more flexible. However, there is a tension: a new layer *may* make the system slower. A classic example is adding a programming language to a system. For example, the EMACS text editor has a core data manipulation library written in C coupled with a Lisp interpreter, through which the library can be used. Much of the editor, and the many extensions people have added to it are written in Lisp, which is arguably much easier than writing them in C and sufficiently fast for text editing.

Currently, there is considerable excitement in the operating systems, networking, and distributed systems communities over what we shall call *resource virtualization*. The basic idea is to add a software layer that provides virtual machines, virtual networks, and even virtual services that are implemented on top of the existing physical resources and services in the network. Because these resources are virtual, we can potentially create a great many of them, make them private to their users, customize them to particular purposes, simplify their administration by making them user- or group-specific, and even inspect them from the *outside* to monitor their performance or detect intrusions.

This course will examine resource virtualization, from the highly influential early work in the 1970s to the present. In particular, it will include:

- Architectural support for virtualization
- Traditional OS-level virtualization
- Paravirtualization
- Virtual servers
- Emulation and binary translation
- Language-level virtualization
- Virtual networking and overlays
- Virtual devices
- Virtual storage and SANs
- Virtual services

- Virtual machine migration
- Remote display
- Virtualization-based computing environments
- Measurement, inference, adaptation and reservation
- Security and virtual machines

Almost all of the readings for the course will be in the form of research papers, with some experience report papers added as well. We will generally read 1-3 papers or equivalent materials for each session, covering fundamental ideas and important recent results. Each paper will be formally presented to the group by a student and then discussed in a round-table manner. A reading list will be available that includes the papers to be read, as well as other related papers.

This is a graduate course and all students in it will be treated like graduate students. I will assume that you are interested in this material, that you can motivate yourself to learn about it, and that you will not be afraid to venture into uncharted territory (i.e., do research). The undergraduate section will differ primarily in that the expectations for the project will be *slightly* lower.

Projects in the previous iteration of the course resulted in several papers published at high quality workshops and conferences.

Project

Over the course of the quarter, you will apply what you learn to a project of your choice, and then document your project in a high quality paper and open presentation. Project topics will be chosen in consultation with me. Projects may be done individually or in groups. Project complexity and expectations will be tied to group size. There are specific projects that I have in mind that would be well-geared to groups of two or three. I will expect weekly project reports.

The expectation for graduate students is that the project will be quality work that the students would not be embarrassed to submit to a workshop. The expectation for undergraduates is that the project be something they would be proud to list on their resumes, although all students are encouraged to aim high. There is a related long-term research project to this course, Virtuoso (more information at http://virtuoso.cs.northwestern.edu/), so there is a potential for projects in this course to turn into longer-term research efforts.

All projects will be presented at a public colloquium.

Example project ideas will be provided in a separate paper handout. Because of the high expectations placed on the project, it is vital that you choose to work on something that interests you deeply and that I can advise strongly.

Exams

There will be no exams

Grading

50 %	Project, **including weekly progress reports**
10 %	Project paper and presentation
20 %	In-class paper presentations of papers
20 %	General classroom participation

Schedule

Thursday, 1/5 Introductory Material
What are virtual machines and what can we do with them?
Reviewing how an operating system works
Required Reading: Collection-Intro; Collection-Smith&Nair; Figueiredo03 (3)

Tuesday, 1/10 Traditional OS-level virtual machines
Goldberg's seminal work
The VM/370 operating system
Required Reading: Goldberg74 (19), Seawright79 (20), Creasy81 (21)

Thursday, 1/12 Architectural support for traditional OS-level VMs
What hardware features are needed
Why it used to be hard to virtualize an IA32 or IA64 machine
Required Reading: Golpek74 (41), Robin00 (43)
PROJECT PROPOSAL DUE

Tuesday, 1/17 Modern architectural support for traditional OS-level VMs
Intel's VT extensions and AMD's Pacifica extensions
Required Reading: Collection-Uhlig, AMD (45)

Thursday, 1/19 Modern VMMs
How something like VMware or Virtual PC works
Required Reading: Collection-Rosenblum; Waldsburger29 (29), Lawton (26)
Other reading: Hall91 (42), Plex/86, Karger91 (25), Virtual 8086 mode

Tuesday, 1/24 Paravirtualization
Blurring the line between VMM and microkernel
Very fast VMMs that you have to port your operating system to
Xen, Denali, UML
Required Reading: Dike00 (35), Collection-Whitaker, Barham03 (38)
Other reading: Hand05 (39)

Thursday, 1/26 Virtual Servers
Making one operating system look like many

Required Reading: Linux V-Server (62), Solaris Zones (64)

Tuesday, 1/31 Emulation and binary translation
Running on a different architecture
SIMICS, QEMU
Required Reading: Mallach73 (46), Mangnusson02 (48), Bellard05 (51)

Thursday, 2/2 Language-level virtual machines
Make your own architecture
UNCOL, JVM, Jit-compilers
Required Reading: Bagley62 (53), Lindholm (55), Adl-Tabatabai (98)

Tuesday, 2/7 Virtual devices
Making peripherals remote, virtual, or both, while keeping them fast
Required Reading: Sugarman01 (65), King03 (66), Cherkasova05 (71)
Other Reading: Menon05 (70)

Thursday, 2/9 Virtual storage and SANs
Putting your disks in someone else's care
Required Reading: Wikipedia (76), Huang04 (75), Beck02 (73)

Tuesday, 2/14 Virtual networking and overlays
Building a new, better network on top of the old network
Required Reading: Chu02 (86), Banerjee03 (89)

Thursday, 2/16 Virtual networking and overlays
Building a new, better network for virtual machines on top of the old network
Required Reading: Sundararaj04 (90), Jiang03 (91), Walfish04 (92)

Tuesday, 2/21 Remote display
Delivering the console to the user
Thin client computing
Required Reading: Richardson98 (120), Baratto05 (124)
Other reading: Lai02 (123), Microsoft00 (122)

Thursday, 2/23 Migration
Delivering the computer to the user
Required Reading: Kozuch02 (113), Clark05 (117), Nelson05 (118)

Tuesday, 2/28 Virtualization-based computing environments
High-end and grid computing
Required Reading: Fraser03 (5), Adablala05 (12), Ruth05 (14)
Other reading: Keahey05 (16), Maccabe05 (17), Shoykhet (9)

Thursday, 3/2 Virtualization-based computing environments
Desktop replacements and servers

Required Reading: Chandra05 (13), Satyanarayanan05 (15)
Other Reading: Garfinkel03 (7), Jiang03 (8), Sapuntzakis03 (6)

Tuesday, 3/7 Measurement, inference, adaptation, and reservation
Automagically making existing, unmodified application run better or cheaper
Required Reading: Sundararaj05 (97), Lange05 (98) Gupta06 (101)

Thursday, 3/9 Security and virtual machines
VMM isolation and codesize arguments
VM introspection
Required Reading: Karger91 (25), Garfinkel03 (108)
Other reading: Dunlop02 (107)

PROJECT DUE

Project Presentations In Finals Week