CS 525: Advanced Distributed Systems

Course Overview
Over the past few decades, the functions of "traditional" operating systems have been scattered out to the edges of distributed systems. Peer-to-peer systems (think: Kazaa), sensor networks, the Grid, PlanetLab, the Internet and the Web are examples. New classes of these systems include datacenters and computing clouds (e.g., EC2, Appengine, Cirrus, Google-IBM cloud). This course focuses on three two case study areas, spanning the classical to bleeding edge: **cloud computing**, **peer-to-peer systems**, and **sensor networks**. We will study efficient protocol design and evaluation, as well as learn high-level system issues. Research in these areas also tends to be scattered across disjoint sets of researchers and conferences. A second emphasis of the course is on bridging the gap between these focus areas and the existing base of **theoretical distributed computing**. This has the potential for applying algorithms or principles from one area to another; the course is an attempt to highlight these possibilities and motivate valuable projects.

Prerequisites
Basic Computer Science and basic computer programming skills are essential. Knowledge of Operating Systems (CS 241 or CS 423), or Networks (CS 438), or Distributed Systems (CS 425), or an equivalent course, or instructor consent, is required.

Course Website `http://www.cs.uiuc.edu/class/cs525`.

Timings
Class: Tuesday and Thursday, 12:30 PM - 1:45 PM, 1131 Siebel Center.
Office Hours (tentative, class days only): Tuesday and Thursday 2 PM - 3 PM, 3112 SC.

Course Staff

Dr. Indranil Gupta
3112 SC, `indy@cs.uiuc.edu`, 265-5517

Course Content
The first few weeks of lectures introduce ground basics in peer to peer systems, theory, sensor networks, and cloud computing systems. Subsequently, over 70 research papers in various areas of distributed systems are presented, discussed, and debated by the students. The selection includes classical and contemporary papers from conferences including, but not limited to, PODC, Middleware, SOSP, OSDI, FAST, NSDI, Usenix, Infocom, SIGCOMM, SASO, etc., as well as ACM and IEEE journal papers. Rather than running through all the papers in a few of such proceedings and journals, we will pick and choose publications appropriate to the stated goals of this course.

An essential component of the course is a project involving at least one non-trivial idea and hands-on implementation. The goal of this project is a conference-quality paper submission and acceptance, and past course projects have done extremely well along this metric. You can collaborate in groups, and I will work with individual groups in defining the project and during its progress. At the end of the course, a top few "best" project papers will be selected, and given

1

special attention for submission to conferences. For the record, 9 out of the 12 papers from the Fall 2003 course have been accepted in conferences (e.g., WCW, ICDCS, MASS and PODC) - a total of 4 journal papers and 8 conference papers came out of that single semester!. Many of the Fall 2004, Spring 2006, Spring 2007, Spring 2008, and Spring 2009 papers are either under conference review or have been accepted into highly competitive conferences (e.g., ICDCS, Infocom, Middleware, MMCN, SASO, QShine, CollaborateCom, GRID, etc.), as well as appear in several top journals (e.g., ACM TAAS, ACM TOSN, IEEE TNSM, JSS, Distributed Computing, etc.).

About the Class
The initial few weeks of class will consist of lecturing, with the intent of building up common knowledge and grounding for the latter half of the course. We will then transition to student-led presentations of papers. Once student-led presentations start, students who are not presenting are expected to write short reviews (1-2 pages total) for any two of the "Main Papers" in that session. Active class participation is required, even in the initial part of the course!

Class Evaluation: Project, review papers, presentation, and class participation. Tentative splits are 50%, 20%, 25%, 5% respectively.

Abridged list of Topics (see course website for more comprehensive list): Distributed computing theory, cloud computing, peer to peer systems, sensor networks, probabilistic algorithms, the Grid, overlays, cloud scheduling, cloud programming, storage, routing, in-network processing, monitoring and management, membership, industrial systems, publish-subscribe, Byzantine fault-tolerance, distributed debugging, green computing, measurement studies, structure of networks.