Programming Languages

Tevfik Koşar

Lecture - I
January 17th, 2006

Contact Information

- Prof. Tevfik Kosar
- Office: 292 Coates
- Phone: 578-9483
- Email: kosar@lsu.edu
- Web: http://www.cct.lsu.edu/~kosar

- Office hours: Tue & Thu, 1:30pm - 2:30pm
- Course web page: http://www.cct.lsu.edu/~kosar/csc4101

- Provide me with your active email address to be added to the class mailing list.
Roadmap

• Meet the Professor
  - Background
  - Teaching philosophy
• Motivation for the Course
  - What expect to learn?
• Introduction to the Course Material
• Administrative details
• Take Photos

Tevfik Kosar

• Joined LSU in August 2005
• Education:
  - PhD: University of Wisconsin-Madison (CS)
  - MS: Rensselaer Polytechnic Institute, NY (CS)
  - BS: Bosporus University, Turkey (CompE)
• Teaching
  - This semester:
    • CSC 4101 - Programming Languages
  - Next year:
    • CSC 4103: Operating Systems
    • CSC 7700: Data Intensive Distributed Computing
Research

- Grid Computing
  - Analogy to the Power Grid
  - A special case for Distributed Computing
  - Spans wide area networks and multiple administrative domains

- The Center for Computation & Technology
  - Spend half of my time there
  - Office: Johnston 333
  - Multi-disciplinary research
  - [http://www.cct.lsu.edu](http://www.cct.lsu.edu)

The Imminent Data “deluge”

- Exponential growth of scientific data
  - 2000: ~0.5 Petabyte
  - 2005: ~10 Petabytes
  - 2010: ~100 Petabytes
  - 2015: ~1000 Petabytes

- “I am terrified by terabytes”
  -- Anonymous

- “I am petrified by petabytes”
  -- Jim Gray

- Moore’s Law outpaced by growth of scientific data!
A High Energy Physics Project: LHC

- The detectors at the LHC will probe fundamental forces in our Universe, such as search for the yet-undetected Higgs Boson.
- Starting in 2006 the LHC accelerator will produce proton-proton collisions with a rate of $10^9$ events/s.
- Four detectors:
  - ATLAS, CMS, ALICE, LHC-B
- LHC Challenges:
  - 11 Petabytes of data per year
  - 100,000 CPUs
  - 5000 physicists, in 300 institutes in 50 countries

A Bioinformatics Project: BLAST

- Goal: decode genetic information and map the genomes of humans, and other species.
- Uses comparative genomics: compares unknown genetic sequences (~billions) to known genomes in search of similarities.
- Current dataset:
  - Several Petabytes
- Future:
  - Exponential Growth: SCARY!
An Educational Technology Project: WCER Educational Video Processing

- Build histories of student learning for use in education research and instruction relying on video data.
- Analyze and share large amount of video.
- 1 hour DV video is ~13 GB
  - A typical educational research video uses 3 cameras => 39 GB for 1 hour
- Current data set:
  - > 500 Terabytes
- Future:
  - Several Petabytes

Astronomy

- Mapping of Universe, detection of new galaxies and stars...

Current Datasets

<table>
<thead>
<tr>
<th>Project</th>
<th>Data Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPOSS</td>
<td>3 TB</td>
</tr>
<tr>
<td>2MASS</td>
<td>12 TB</td>
</tr>
<tr>
<td>SDSS</td>
<td>40 TB</td>
</tr>
</tbody>
</table>

Future Productions

<table>
<thead>
<tr>
<th>Project</th>
<th>Data Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>WFCAM</td>
<td>20 TB/year</td>
</tr>
<tr>
<td>VISTA</td>
<td>100 TB/year</td>
</tr>
<tr>
<td>LSST</td>
<td>1000 TB/year</td>
</tr>
</tbody>
</table>
How to access and process distributed data?

Interested?

- Send an email to kosar@lsu.edu
- Register for a independent study (CSC 4999)
- Take 3 credits for doing research in one of these interesting topics during one semester
Teaching Philosophy

• Goal:
  - For instructor: teaching the material
  - For student: learning and applying the material in real life
• Grades are of second degree importance
• Do not memorize, understand the material
  - Exams may be openbook!
• You are only responsible from material
  - Covered in the class
  - Part of projects or homework assignments

Programming Languages

• How many different programming languages are there?
  - More than 200!
• Can you name some of them?
• Which ones have you used before?
  - Java
  - C++
  - C
  - Lisp/Scheme
  - Prolog
  - ....
Language of the Computer

- Machine Language
  - Consists of 0’s and 1’s
  - Which refers to high and low voltage states
  - 0010 0111 1010 1101 1111 1111 1101 0000 ....
  - 27bdf0d0 afb0014 0c1002a8 ...
- Assembly Language
  - push bx
  - mov bx
  - div bx
  - add dx
  - Direct mapping to machine language
- Higher Level Languages
  - C, C++, Java, Pascal, Scheme, Prolog ..
  - First one: Fortran

Why are there so many programming languages?

- Special Purposes
  - Each language is designed to solve a certain problem:
    - Perl for string parsing and manipulation
    - C/C++ for systems programming
    - Java for platform independent programs
    - Prolog for logic programming and AI
    - Fortran for numerical computations
- Personal Preferences
- Evolution
  - Learn better ways of doing things over time..
  - eg. from “go to” to “while” loops, “case” statements
What makes a language successful?

- easy to learn (BASIC, Pascal, LOGO, Scheme)
- easy to express things, easy use once fluent, “powerful” (C, Common Lisp, APL, Algol-68, Perl)
- easy to implement (BASIC, Forth)
- possible to compile to very good (fast/small) code (Fortran)
- backing of a powerful sponsor (COBOL, PL/1, Ada, Visual Basic)
- wide dissemination at minimal cost (Pascal, Turing, Java)

Programming Paradigms

- Group languages as
  - declarative
    - functional (Scheme, ML, pure Lisp, FP)
    - logic, constraint-based (Prolog, VisiCalc, RPG)
  - imperative
    - von Neumann (Fortran, Pascal, Basic, C)
    - object-oriented (Smalltalk, Eiffel, C++)
    - scripting languages (Perl, Python, JavaScript, PHP)
Why study programming languages?

- Help you choose a language
- Make it easier to learn new languages
  - Syntactic similarities
    - C++ vs Java
  - Conceptual similarities
    - C vs Pascal
- Help you make better use of whatever language you use
  - Choose among alternative ways
    - Using arrays vs pointers
    - Loops vs Recursion
  - Simulate useful features in languages that lack them
    - Faking pointers
    - Faking modularity

Textbooks

- Required text:
  - Programming Language Pragmatics (2nd edition)
  - by Michael Scott, Morgan Kauffman Publishers, 2005

- Recommended text:
  - Concepts of Programming Languages (6th edition)
  - Robert W. Sebesta, Addison-Wesley, 2003

- There will be additional links for supplementary course material on the course web page
Grading

• The end-of-semester grades will be composed of:
  - Popup Quizzes : 5%
  - Active Contribution : 5%
  - Homework : 15%
  - Projects : 30%
  - Midterm : 20%
  - Final : 25%

Reading Assignment

• Read chapter 1 from Programming Language Pragmatics (PLP).
Any Questions?

Hmm..