CS 744: BIG DATA SYSTEMS

Shivaram Venkataraman
Fall 2018
WHO AM I?

New faculty in Computer Science!

PhD Thesis at UC Berkeley:
System Design for Large Scale Machine Learning

Industry: Google, Microsoft Research
Open source: Apache Spark committer
CALL ME

Shivaram or Prof. Shivaram
OUTLINE

- What is this course about?
- Goals
- Class format
- Next Steps
BRIEF HISTORY OF BIG DATA
GOOGLE 1997
“…Storage space must be used efficiently to store indices and, optionally, the documents themselves. The indexing system must process hundreds of gigabytes of data efficiently…”

The Anatomy of a Large-Scale Hypertextual Web Search Engine

Sergey Brin and Lawrence Page
GOOGLE 2001

Commodity CPUs

Lots of disks

Low bandwidth network

Cheap !
Facebook’s daily logs: 60 TB

Google web index: 10+ PB
“scientific breakthroughs will be powered by advanced computing capabilities that help researchers manipulate and explore massive datasets”

-- Jim Gray
SCIENTIFIC APPLICATIONS
Solar Flare Prediction Using Photospheric and Coronal Image Data.

~ 2 PB

Working with data from Solar Dynamics Observatory
[Brown et. al SDO Primer 2010]

Solar Flare Prediction Using Photospheric and Coronal Image Data.
[Jonas et. al American Geophysical Union, 2016]
Data Growth is Outpacing Computing Growth

Graph based on average growth

Source: More Data, More Science and... Moore’s Law  [Kathy Yellick]
Google data centers in The Dalles, Oregon
DATACENTER EVOLUTION

Capacity:
~10000 machines

Bandwidth:
12-24 disks per node

Latency:
256GB RAM cache
Above the Clouds: A Berkeley View of Cloud Computing

Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia
(Comments should be addressed to abovetheclouds@cs.berkeley.edu)

UC Berkeley Reliable Adaptive Distributed Systems Laboratory *
http://radlab.cs.berkeley.edu/

“…long-held dream of computing as a utility…”
FROM MID 2006

Rent virtual computers in the “Cloud”

On-demand machines, spot pricing
<table>
<thead>
<tr>
<th>Machine</th>
<th>Memory (GB)</th>
<th>Compute Units (ECU)</th>
<th>Local Storage (GB)</th>
<th>Cost / hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1.micro</td>
<td>0.615</td>
<td>1</td>
<td>0</td>
<td>$0.02</td>
</tr>
<tr>
<td>m1.xlarge</td>
<td>15</td>
<td>8</td>
<td>1680</td>
<td>$0.48</td>
</tr>
<tr>
<td>cc2.8xlarge</td>
<td>60.5</td>
<td>88 (Xeon 2670)</td>
<td>3360</td>
<td>$2.40</td>
</tr>
</tbody>
</table>

1 ECU = CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor
<table>
<thead>
<tr>
<th>Machine</th>
<th>Memory (GB)</th>
<th>Compute Units (ECU)</th>
<th>Local Storage (GB)</th>
<th>Cost / hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>t2.micro</td>
<td>0.615</td>
<td>1</td>
<td>0</td>
<td>$0.013</td>
</tr>
<tr>
<td>r3.xlarge</td>
<td>15</td>
<td>8-13</td>
<td>1680 80(SSD)</td>
<td>$0.35</td>
</tr>
<tr>
<td>r3.8xlarge</td>
<td>60.5-244</td>
<td>88 104 (Ivy Bridge)</td>
<td>3360 640(SSD)</td>
<td>$2.80</td>
</tr>
</tbody>
</table>

1 ECU = CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor
<table>
<thead>
<tr>
<th>Machine</th>
<th>Memory (GB)</th>
<th>Compute Units (ECU)</th>
<th>Local Storage (GB)</th>
<th>Cost / hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>t2.nano</td>
<td>0.5</td>
<td>1</td>
<td>0</td>
<td>$0.006</td>
</tr>
<tr>
<td>t2.micro</td>
<td>0.615</td>
<td>1</td>
<td>0</td>
<td>$0.013</td>
</tr>
<tr>
<td>r3.8xlarge</td>
<td>60.5–244</td>
<td>88 104 (Ivy Bridge)</td>
<td>3360 640(SSD)</td>
<td>$2.80</td>
</tr>
<tr>
<td>x1 (TBA)</td>
<td>2 TB</td>
<td>4 * Xeon E7</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Machine</td>
<td>Memory (GB)</td>
<td>Compute Units (ECU)</td>
<td>Local Storage (GB)</td>
<td>Cost / hour</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>---------------------</td>
<td>--------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>t2.nano</td>
<td>0.5</td>
<td>1</td>
<td>0</td>
<td>$0.006</td>
</tr>
<tr>
<td>r3.8xlarge</td>
<td>60.5–244</td>
<td>88 104 (Ivy Bridge)</td>
<td>3360 640 (SSD)</td>
<td>$2.66</td>
</tr>
<tr>
<td>x1.32xlarge</td>
<td>2 TB</td>
<td>4 * Xeon E7</td>
<td>3.4 TB (SSD)</td>
<td>$13.338</td>
</tr>
<tr>
<td>p2.16xlarge</td>
<td>732 GB</td>
<td>16 Nvidia K80 GPUs</td>
<td>0</td>
<td>$14.4</td>
</tr>
</tbody>
</table>
### AMazon EC2 (2018)

<table>
<thead>
<tr>
<th>Machine</th>
<th>Memory (GB)</th>
<th>Compute Units (ECU)</th>
<th>Local Storage (GB)</th>
<th>Cost / hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>t2.nano</td>
<td>0.5</td>
<td>1</td>
<td>0</td>
<td>$0.0058</td>
</tr>
<tr>
<td>r5d.24xlarge</td>
<td>244 768</td>
<td>104-96</td>
<td>4x900 NVMe</td>
<td>$6.912</td>
</tr>
<tr>
<td>x1.32xlarge</td>
<td>2 TB</td>
<td>4 * Xeon E7</td>
<td>3.4 TB (SSD)</td>
<td>$13.338</td>
</tr>
<tr>
<td>p3.16xlarge</td>
<td>488 GB</td>
<td>8 Nvidia Tesla V100 GPUs</td>
<td>0</td>
<td>$24.48</td>
</tr>
</tbody>
</table>
The Joys of Real Hardware

Typical first year for a new cluster:

~0.5 overheating (power down most machines in <5 mins, ~1-2 days to recover)
~1 PDU failure (~500-1000 machines suddenly disappear, ~6 hours to come back)
~1 rack-move (plenty of warning, ~500-1000 machines powered down, ~6 hours)
~1 network rewiring (rolling ~5% of machines down over 2-day span)
~20 rack failures (40-80 machines instantly disappear, 1-6 hours to get back)
~5 racks go wonky (40-80 machines see 50% packetloss)
~8 network maintenances (4 might cause ~30-minute random connectivity losses)
~12 router reloads (takes out DNS and external vips for a couple minutes)
~3 router failures (have to immediately pull traffic for an hour)
~dozens of minor 30-second blips for dns
~1000 individual machine failures
~thousands of hard drive failures
slow disks, bad memory, misconfigured machines, flaky machines, etc.

Long distance links: wild dogs, sharks, dead horses, drunken hunters, etc.
How do we program this?
BIG DATA SYSTEMS
Scalable Storage Systems

Datacenter Architecture

Computational Engines

Resource Management

Applications

Machine Learning

SQL

Streaming

Graph

TensorFlow

Spark

hadoop

kubernetes

Open Compute Project
GOALS

1. Understand system design aspects
2. Explain, discuss research contributions
3. Expertise to deploy, use and extend systems
4. Perform new research and implementation

Grading breakdown in course website
GRADING

• Paper reviews: 10%
• Class Participation and Presentation: 15%
• Assignments (in groups): 20% (2 @ 10% each)
• Midterm exam: 20%
• Final Project (in groups): 35%
LECTURE FORMAT

3 papers per class: 1 Main paper, 2 optional papers
Schedule http://cs.wisc.edu/~shivaram/cs744-fa18

Required: Reading the main paper and writing a review
Review on Piazza by 9:00 am on day of class
PAPER REVIEW FORMAT

Less than one page!

- One or two sentence summary of the paper
- Description of the problem
- Summary of the contributions
- One flaw or thing that can be improved
- One thing you were confused about
CLASS PRESENTATIONS

Part 1
- First 20 min: Main paper presented by instructor
- Clarify questions posted on Piazza

Part 2, 3
- 20-25 min talks presented by students
- Compare and relate to main paper
- Email slides to staff by 9am the day before
CLASS PRESENTATION FORMAT

1. Problem: What is the paper trying to solve? How real is it?
2. Key idea: What is the main idea in the solution?
3. Novelty: What is different from previous work, and why?
4. Critique: Is there anything you would change in the solution?
5. Comparison: How does this paper relate to the main paper?
ASSIGNMENTS

Two homework assignments using NSF CloudLab
- Assignment 0: Setup CloudLab account
- Assignment 1: Data Processing/Spark
- Assignment 2: Machine Learning/Tensorflow

Short coding based assignments. Preparation for course project
Work in groups of three
COURSE PROJECT

Main grading component in the course!

Goal: Explore new research ideas or significant implementation in the area of Big Data systems

Research: Work towards workshop/conference paper
Implementation: Work towards open source contribution
COURSE PROJECT EXAMPLES

Example: Research
How do we scheduling distributed machine learning jobs while accounting for performance, efficiency, convergence?

Example: Implementation-heavy
Implement a new module in Apache YARN that allows GPUs to be allocated to machine learning jobs.
Project Selection:
- List of course project ideas will be posted by Tuesday 9/11
- Form groups of three
- Come up with a short list of ideas or propose your own!
- Meeting with instructors to finalize project (around 9/20)

Grading:
- Mid-term write up
- Final project report
COURSE LOGISTICS

Instructor office hours: Tue Thu 2-3PM at 7367 CS

TA office hours: MW 9-10AM at 4244 CS

Discussion, Questions: Use Piazza!
WAITLIST

- Class size is limited to 45 for this semester
- Focus on research projects, class presentations, discussion
- Course will be taught in Spring 2019

If you are enrolled but don’t want to take, please drop ASAP!
If you are on the waitlist: Fill out
https://goo.gl/forms/UrtHMj7WUMkoo7E53
CAN I AUDIT THE COURSE?

- Audit students are welcome!
- Review papers on Piazza
- Do assignments on CloudLab

- Not enough slots for presentation or course projects
BEFORE NEXT CLASS

Join Piazza: https://piazza.com/wisc/fall2018/cs744

Presentation Preference
https://goo.gl/forms/XrZNMe4q4p8yBUzX2

Project/Assignment Groups
https://goo.gl/forms/cB532EWEfFmSUtl52